Copula Directed Acyclic Graphs

Eugen Pircalabelu · Gerda Claeskens · Irène Gijbels

June 26, 2015

Abstract A new methodology for selecting a Bayesian network for continuous data outside the widely used class of multivariate normal distributions is developed. The 'copula DAGs' combine directed acyclic graphs and their associated probability models with copula C/D-vines. Bivariate copula densities introduce flexibility in the joint distributions of pairs of nodes in the network. An information criterion is studied for graph selection tailored to the joint modeling of data based on graphs and copulas. Examples and simulation studies show the flexibility and properties of the method.

Keywords Directed acyclic graph · Copula · C-vine · D-vine · Model selection

1 Introduction

In recent decades there has been a fast increase in the ease with which multivariate data are gathered. General multivariate techniques have been developed to deal with such data. One class of techniques for analyzing and interpreting multivariate data is based on graphical models (see, e.g., Lauritzen, 1996; Cox and Wermuth, 1996). By representing a random variable as a node, a graphical model depicts relations between such variables, either in an associative or causal form, by drawing edges between the nodes. Allowing edges to be of different types (directed, undirected, bi-directed) defines different types of graphs with corresponding interpretations for relations between the components of the associated random vector.

In this paper we focus on Bayesian networks (see, e.g., Heckerman and Geiger, 1995; Spirtes et al, 2000; Koller and Friedman, 2009), that comprise first the graphical representation in the form of a directed acyclic graph (DAG) for which all the edges between the nodes have a single direction and no loops are allowed, and second the decomposition of a joint probability density function as a product of conditional and marginal density functions according to the graph. For example, the absence of edges between two nodes can be interpreted as a conditional independence between the variables associated with those nodes conditionally on other random variables.

E. Pircalabelu, G. Claeskens
ORSTAT and Leuven Statistics Research Center
KU Leuven
Naamsestraat 69, 3000 Leuven, Belgium
E-mail: eugen.pircalabelu@kuleuven.be
E-mail: gerda.claeskens@kuleuven.be

I.Gijbels
Department of Mathematics and Leuven Statistics Research Center
KU Leuven,
Celestijnenlaan 200B, 3001 Leuven (Heverlee), Belgium
E-mail: irene.gijbels@wis.kuleuven.be

The main contribution of the paper is to present a new methodology for selecting a Bayesian network for continuous data outside the widely used class of multivariate normal distributions. We introduce flexibility in the distribution by using bivariate copula densities to model connections between pairs of nodes in the network and present a new score criterion for graph selection tailored to the joint modeling of data based on graphs and copulas. In order to deal with the high-dimensional aspect of the data, the C- and D-vines, short for *c*anonical and *d*rawable vines (Bedford and Cooke, 2001), play a central role, due to the fact that these decompositions of a multivariate density employ a series of bivariate conditional and unconditional copulas to represent a general multivariate density.

Unlike social networks where connections are observed (e.g., who is sending text messages to whom), in probabilistic graphical models a random vector (or a sample of vectors all from the same underlying multivariate probability distribution) is observed. A statistical analysis will try to discover the relations between the components of the multivariate vectors. In other words, one aims to discover the structure of the graph e.g., where to draw edges in the graphical representation. The goal in graphical modeling is thus, to estimate a plausible decomposition of a general multivariate density, that can be linked visually to a graphical object, in which certain simplifying assumptions of marginal and conditional independencies are made.

When using Bayesian networks (BN), one is faced with two possibilities: either use external expert advice and put forward a plausible model, or estimate such a plausible structure from the data. When one is working with continuous data, most often one uses models that, for ease of computational burden and efficient algorithmic implementations, rely on the multivariate normality assumption. In this paper we allow for other continuous distributions via copula models. The estimation of the graph structure is usually done either by a *testing* procedure, where one is trying to discover conditional independencies using formal hypothesis tests, or by using a *scoring* procedure where one selects the graph that optimizes a certain score. We start by presenting a simple example for which two estimated graphs are presented. The dataset used for illustration, is a subset of the 'Wine' dataset that comes from the UCI Machine Learning repository. It contains 178 sample cases (different wines) and 11 chemical measurements among which alcohol, malic acid, magnesium content or color and hue. We refer to Bache and Lichman (2013) for more information about the dataset.

In Figure 1 we present two estimated DAGs, one based on the PC algorithm (Spirtes et al, 2000) which uses hypothesis testing for discovering edges in the graph and one based on a Bayesian Gaussian equivalent scoring criterion abbreviated by 'BGe' (Heckerman and Geiger, 1995). Both approaches assume multivariate normality. The immediate observation one can make, is that the BGe based graph estimates more edges than the PC based graph and this impacts the conditional independencies that can be read from the graph. For example, in the PC graph one reads that the alcohol level (Alch) of a wine is independent of the proanthocyanins level (Prnt) if one conditions upon the flavanoids level (Flvn). It is thus crucial, that one models the data in an appropriate manner to reveal a graph that is most informative, as different graphs might lead to different assumptions being made about the underlying data generating process.

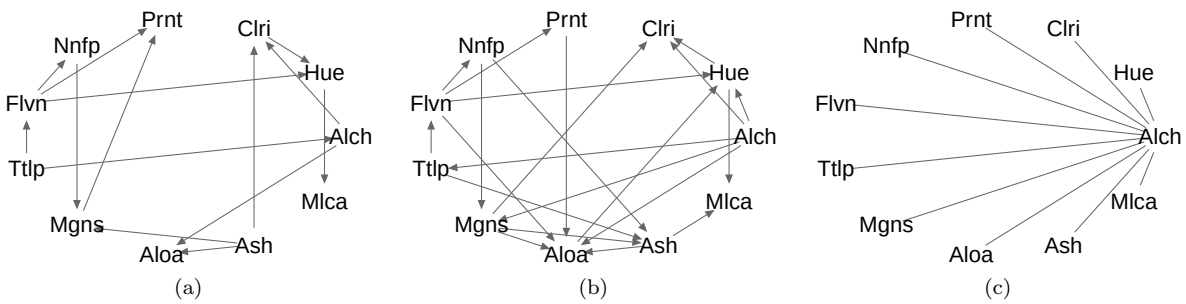


Fig. 1: Wine data. Estimated DAGs using (a) the PC($\alpha = .1$) algorithm and (b) the BGe score, (c) the first tree of a C-vine using Alch as central node.

Another quite different technique for analyzing multivariate dependencies is by using copulas (Nelsen, 2006; Mari and Kotz, 2001), which are joint distribution functions evaluated at the values corresponding to the marginal distribution functions. The key aspect distinguishing copulas from Bayesian networks is that, when modeling data one is generally focusing on the probabilistic aspect and not considering the additional information provided by visual aids such as a graph. Vine copula models (Bedford and Cooke, 2001) are multivariate copulas using bivariate copulas as building blocks and can be represented by using graphical objects in which edges connecting two nodes have the explicit meaning of presenting which two variables in the graph are modeled together using a copula distribution. Figure 1(c) presents the first tree of a C-vine construction where the alcohol level (Alch) is used as a central node connected to all other nodes.

As further explained in Section 2, the edges in these graphical objects serve two different purposes. In the DAG they represent to a large extent an interventional aspect of the type 'X determines Y' ($X \to Y$), whereas in the vine graph they show bivariate dependencies of the type 'X and Y are dependent and modeled bivariately by a copula' ($X - Y$). One other important difference is that the vine graphs have a hierarchical structure, as there is not a unique graph as a result of the procedure, but several graphs, see Figure 2, all hierarchically linked in the sense that as we move down in the hierarchy, bivariate data are modeled conditioning on larger and larger sets.

Our proposed procedure combines directed acyclic graphs and their associated probability model with copula C/D-vines in order to construct 'copula based DAGs', or short 'cDAGs'. We exploit certain connections and similarities that exist between these two statistical techniques with the explicit purpose of estimating a graphical model, a network, for continuous data that are not necessarily normally distributed. The approach we use is a score based learning scheme, where one modifies an initial graph based on improvements in the score, until a local optimum score is reached. For this purpose, given a collection of copula families, we construct a nodewise decomposable score based on a series of implied C/D-vine decompositions which can be used to select both the graph and the copulas that nodewise optimize the score.

To researchers active in the copula field, our approach brings in an estimated DAG that shows causal paths between variables while exploiting low dimensional copulas, while for researchers using Bayesian networks the methodology offers flexibility in modeling dependencies between nodes allowing a wide range of continuous distributions. Both the construction of the graph as well as the copula family selection are incorporated into a novel information criterion, for which we investigate some theoretical properties.

In the literature, several other procedures linking copulas to graphical models are encountered. Elidan (2010, 2012) parametrizes the conditional density of a node given its parents in terms of higher dimensional copulas, while Bauer et al (2012) use a 'pairwise copula construction' (PCC) for the entire joint density with the additional need to specify an ordering of the nodes (which node is allowed to be an ancestor of other nodes) and this is generally hard to specify in practice. In Harris and Drton (2013) the constraint based PC algorithm is used to estimate DAGs using rank-based measures of association for a Gaussian copula. Liu et al (2009) use copulas to estimate sparse high dimensional undirected graphs. Kurowicka and Cooke (2002) show how using an elliptical copula one can associate a Bayesian net to a vine construction or vice versa. Hanea et al (2010) and Hanea (2011) construct non-parametric belief networks using D-vine decompositions to represent the conditional independencies in the graph. In contrast, our approach has the advantage of allowing more flexibility since we model the parent-to-child edges using C-vine models and the parent-to-parent edges with D-vines. The approach used in Hanea (2011) cleverly maps the conditional independencies that hold in the directed graph to a D-vine which in turn forces some conditional correlations to be zero, whereas our approach allows for full local vine decompositions. Our main starting point is represented by the undirected connections in the moralized graph which dictate which nodes are involved in the copula decompositions. Several other search procedures require formal hypotheses testing, which requires care to avoid accumulating probabilities of type-I errors, or need to specify a threshold, e.g. for the size of a conditional correlation coefficient in order to decide on the inclusion of an edge. Our approach uses an information criterion to select the final model which is used to select both the directed structure and the best fitting copulas.

Stepping outside the multivariate normality assumption for graphical modeling is mainly addressed for undirected graphical models, see for example, Wainwright and Jordan (2008), Jalali et al (2010),

Yang et al (2012), Lee and Hastie (2014), or Loh and Wainwright (2013) to name just a few, where most often one models each node assuming its distribution is a member of the more general exponential family.

The outline of the paper is as follows: Section 2 provides an introduction to both Bayesian networks and copula modeling. Section 3 presents the main ideas based on which we combine the two statistical approaches. A motivating example is presented in Section 4, while Sections 5 and 6 detail the selection of the network and copula families at both a theoretical and algorithmic level. Sections 7 and 8 evaluate the method empirically using simulated data and the 'Euro Stoxx 50' financial dataset. Section 9 concludes.

2 Background information on Bayesian networks and copulas

Some of the main concepts involved in modeling multivariate data are briefly revised, first based on a Bayesian network and second on a copula approach.

2.1 Bayesian networks

Bayesian networks represent an important class of graphical models with wide applications in many different fields ranging from biomedical studies (Lucas, 2007), transportation (Madsen and Kjærulff, 2007) and aeronautics (Morales Nápoles, 2010) to language processing (Peshkin et al, 2003) among others. Their popularity rests upon a relative ease of interpretation and implementation of such models, accompanied by a solid mathematical and statistical theoretical framework. In a multivariate context, the starting point of such graphical models is a basic factorization property of joint density (or probability mass) functions. For a random vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$ of length p coming from a multivariate density denoted as $f(x_1, x_2, \ldots, x_p)$, based on the chain rule property of densities the following decomposition holds true:

$$f(x_1, \ldots, x_p) = f_1(x_1) f_{2|1}(x_2|x_1) f_{3|1,2}(x_3|x_1, x_2) \cdot \ldots \cdot f_{p|1,2,\ldots,p-1}(x_p|x_1, x_2, \ldots, x_{p-1}), \tag{2.1}$$

where for example, the function $f_{p|1,2,\ldots,p-1}(x_p|x_1, x_2, \ldots, x_{p-1})$ is used to denote the conditional density of X_p when the conditioning set comprises all variables in the set $\{X_1, X_2, \ldots, X_{p-1}\}$, and only those. Analogous definitions hold for all other conditional densities in (2.1). The marginal density of a variable X_l is denoted throughout the paper as $f_l(x_l)$.

With large p, the last factors in the product (2.1) are quite cumbersome, in the sense that they contain many variables in the conditioning set. For a statistical modeler, this phenomenon alludes to a variable selection problem, namely whether all of the variables are actually needed in the conditioning sets in order to get an accurate construction of the joint density. For example, when X_p is conditionally independent of say $\{X_1, X_2, \ldots, X_{p-3}\}$ given the variables $\{X_{p-2}, X_{p-1}\}$ (we write it as $X_p \perp X_1, X_2, \ldots X_{p-3}|X_{p-2}, X_{p-1}$) the last factor in (2.1) can be replaced by a more parsimonious conditional density since

$$f_{p|1,2,\ldots,p-1}(x_p|x_1, x_2, \ldots, x_{p-2}, x_{p-1}) = f_{p|p-2,p-1}(x_p|x_{p-2}, x_{p-1})$$

and thus, knowing such independencies is beneficial for modeling purposes when representing the joint density $f(x_1, \ldots, x_p)$.

To state it concisely, the main objective is to re-specify joint density functions of multivariate random variables as functions of densities that involve conditioning on only a small number of variables, which is equivalent to specifying and assuming a number of conditional independencies.

Let $G(E, V)$ be a graph based on a set of nodes (V), a set of edges (E), and a set of random variables $\{X_i : i \in V\}$. Each of the variables $X_1, \ldots, X_p$ has a corresponding node in the set $V = \{1, \ldots, p\}$ and the set of edges E is a subset of $V \times V$, the set of ordered pairs of distinct nodes. A directed edge $i \to j$ in E is denoted by (i, j) and we refer to node i (or variable X_i) as a *parent* of node j (or variable X_j), while node j is referred to as a *child* of node i. A directed path between nodes i and z is a sequence of nodes that starts in i and by following the directionality of the arrows leads to node z (e.g. $i \to j \to \ldots \to y \to z$).

Node i is said to be an *ancestor* of z if there exists such a directed path between the two nodes, or if $i = z$. Node z is referred to as a *descendant* of i.

BNs are defined as a class of statistical models, consisting of a graph $G(E, V)$ and a probability density f, with two particular characteristics. First, the graph contains only directed edges between pairs of nodes, such that there are no feedback loops (referred to as the 'acyclicity' property). That is, any directed path starting at node i cannot lead back to i. Second, the joint multivariate probability density function (pdf) of $(X_1, \ldots, X_p)$ factorizes as

$$f(x_1, \ldots, x_p) = \prod_{l=1}^{p} f_{l|pa(l)}(x_l|pa(x_l)), \tag{2.2}$$

where the conditioning is on $pa(x_l)$, the set of parental variables of X_l (see Lauritzen, 1996). Graphically, this is represented by a directed arrow from each of the 'parents' to the 'children'.

We further say that f has the local Markov property with respect to G, or equivalently f decomposes according to G, if

$$\text{for all } l \in V, \quad l \perp nd(l)|pa(l)$$

where the symbol $\perp$ denotes independence and $nd(l)$ denotes the set of non-descendants (excluding the parents) of node l. That is, any node is independent of its non-descendants when conditioned on its corresponding parents. Since we create a one-to-one correspondence between each variable X_i and a node in the graph G, in the remaining parts of the article we will sometimes use the two terms interchangeably.

2.2 Copulas and vines

A copula C is a multivariate distribution for which all marginal distributions are uniform on the interval $[0, 1]$. More formally, let $\{U_1, \ldots, U_p\}$ be a set of p random variables uniformly distributed on $[0, 1]$. Then a copula function $C : [0, 1]^p \to [0, 1]$ is a joint distribution function such that

$$C(u_1, \ldots, u_p) = P(U_1 \leq u_1, \ldots, U_p \leq u_p).$$

Most importantly, Sklar (1959) proved that for any multivariate distribution $F(x_1, \ldots, x_p)$ there exists a copula function C such that

$$F(x_1, \ldots, x_p) = C(F_1(x_1), \ldots, F_p(x_p))$$

meaning that every joint distribution can be obtained from the marginal distributions F_j, $j = 1, \ldots, p$ through the copula function.

For absolutely continuous random variables the quantile function is uniquely defined. Hence, since $C(u_1, \ldots, u_p) = F(F_1^{-1}(u_1), \ldots, F_p^{-1}(u_p))$ differentiating this expression with respect to the marginal distribution leads to the copula density expression, assumed to exist,

$$c(u_1, \ldots, u_p) = \frac{f(F_1^{-1}(u_1), \ldots, F_p^{-1}(u_p))}{\prod_{l=1}^{p} f_l(F_l^{-1}(u_l))} \Leftrightarrow c(F_1(x_1), \ldots, F_p(x_p)) = \frac{f(x_1, \ldots, x_p)}{\prod_{l=1}^{p} f_l(x_l)}. \tag{2.3}$$

Bedford and Cooke (2001, 2002) based on Joe (1996) as well as Kurowicka and Cooke (2006) studied the 'vine' as a general graphical model to describe how multivariate copulas can be reconstructed from simpler bivariate copulas (also referred to as 'pair copulas'). The 'pair-copula constructions' decompose multivariate probability densities into a product of bivariate copulas, where one copula can be chosen independently from any other bivariate copula involved, which offers the advantage that high-dimensional multivariate problems can be tackled through bivariate modeling.

Aas et al (2009) and Czado (2010) described statistical frequentist estimation and inference techniques for what is now known as C- and D-vines, while Bayesian approaches can be found in Min and Czado (2010, 2011), Smith et al (2010) and Czado et al (2011). Another development are the 'regular' vines, referred to as R-vines, of which the C- and D-vines are subclasses, see Brechmann et al (2012) and Dißmann et al (2013).

By $c_{i,j}\{F_i(x_i), F_j(x_j)\}$ we denote the unconditional pair-copula density corresponding to variables X_i and X_j evaluated at the marginal cumulative distribution functions $F_i(x_i)$ and $F_j(x_j)$ (with marginal density functions f_i and f_j). The function

$$c_{j,j+i|1,\ldots,j-1}\{F_{j|1,\ldots,j-1}(x_j|x_1,\ldots,x_{j-1}), F_{j+i|1,\ldots,j-1}(x_{j+i}|x_1,\ldots,x_{j-1})\} \tag{2.4}$$

denotes the pair copula corresponding to variables X_j and X_{j+i} conditioned on the set of variables $\{X_1,\ldots,X_{j-1}\}$. Similarly, the function

$$c_{i,i+j|i+1,\ldots,i+j-1}\{F_{i|i+1,\ldots,i+j-1}(x_i|x_{i+1},\ldots,x_{i+j-1}), F_{i+j|i+1\ldots i+j-1}(x_{i+j}|x_{i+1},\ldots,x_{i+j-1})\} \tag{2.5}$$

is used to denote the pair copula corresponding to variables X_i and X_{i+j} conditioned on the set of variables $\{X_{i+1},\ldots,X_{i+j-1}\}$. When one of the indices is zero, the conditioning set is defined to be the empty set, and conditioning on an empty set is understood to be the same as no conditioning. A conditional copula reflects interest in modeling the dependence structure of a bivariate vector given the variables in the conditioning set and knowing whether this relationship changes as a function of that set. We refer to Gijbels et al (2011) for an exposition on conditional bivariate copulas and associated conditional dependence measures.

The functions (2.4) and (2.5) play a crucial role in this paper, because they are the building blocks on which a C-vine and a D-vine representation of a joint density function is constructed. Based on a C-vine representation the joint density $f(x_1,\ldots,x_p)$ can be decomposed as

$$\prod_{l=1}^{p} f_l(x_l) \prod_{j=1}^{p-1}\prod_{i=1}^{p-j} c_{j,j+i|1,\ldots,j-1}\{F_{j|1,\ldots,j-1}(x_j|x_1,\ldots,x_{j-1}), F_{j+i|1,\ldots,j-1}(x_{j+i}|x_1,\ldots,x_{j-1})\}. \tag{2.6}$$

To better understand such a decomposition, Figure 2 gives a graphical representation of the underlying connections between the variables. For brevity of exposition, in the following whenever we speak of node '1', '2', etc. all statistical statements refer to the variables that are associated with the nodes and the undirected edges are used to denote the fact that two nodes i and j are coupled in a conditional (or unconditional) copula function $c_{i,j}$. We use the convention that if the set $\{i, i+j\}$ and the conditioning set $\{i+1,\ldots,i+j-1\}$ contain the same elements (which happens in both the C-vine and D-vine case only at the first level, when $j=1$) then one is using the unconditional copula function $c_{i,i+j}$.

A C-vine starts with one central node which in Figure 2 is the node with label '1'. All other nodes (2 to 6 in this example) connect to the central node in a tree structure. Inspecting the above C-vine decomposition, this situation corresponds to setting the index $j=1$ and using the convention that if $j-1<1$ then we are actually not conditioning on any variables. This means that at the first level we are using in the decomposition in (2.6), the following unconditional bivariate copulas: $\{c_{1,2}, c_{1,3}, c_{1,4}, c_{1,5}, c_{1,6}\}$. At the second level ($j=2$) we combine the nodes that were connected at the previous level, in this case '12', '13', ..., '16' which now become the new nodes of the tree. This corresponds to using the conditional copula functions $\{c_{2,3|1}, c_{2,4|1}, c_{2,5|1}, c_{2,6|1}\}$. At the third level ($j=3$) one uses the set $\{c_{3,4|1,2}, c_{3,5|1,2}, c_{3,6|1,2}\}$ and the same procedure is repeated until one reaches level $j=5$, which involves using only one conditional copula, namely $c_{5,6|1,2,3,4}$. In order then to obtain the value of the joint density one proceeds by multiplying all the bivariate copulas that were specified at all levels, as in (2.6).

In a roughly similar manner, a decomposition of the same density $f(x_1,\ldots,x_p)$ based on a D-vine representation is

$$\prod_{l=1}^{p} f_l(x_l) \prod_{j=1}^{p-1}\prod_{i=1}^{p-j} c_{i,i+j|i+1,\ldots,i+j-1}\{F_{i|i+1,\ldots,i+j-1}(x_i|x_{i+1},\ldots,x_{i+j-1}), F_{i+j|i+1\ldots i+j-1}(x_{i+j}|x_{i+1},\ldots,x_{i+j-1})\}. \tag{2.7}$$

Figure 3 graphically depicts the structure underlying a D-vine on 6 nodes. Now, all the nodes are linked sequentially to each other to form a 'chain'-like structure. At the first level ($j=1$) one is using the bivariate unconditional copulas $\{c_{1,2}, c_{2,3}, c_{3,4}, c_{4,5}, c_{5,6}\}$. At the second level ($j=2$), one is using the following conditional copula functions $\{c_{1,3|2}, c_{2,4|3}, c_{3,5|4}, c_{4,6|5}\}$ and the remaining bivariate conditional

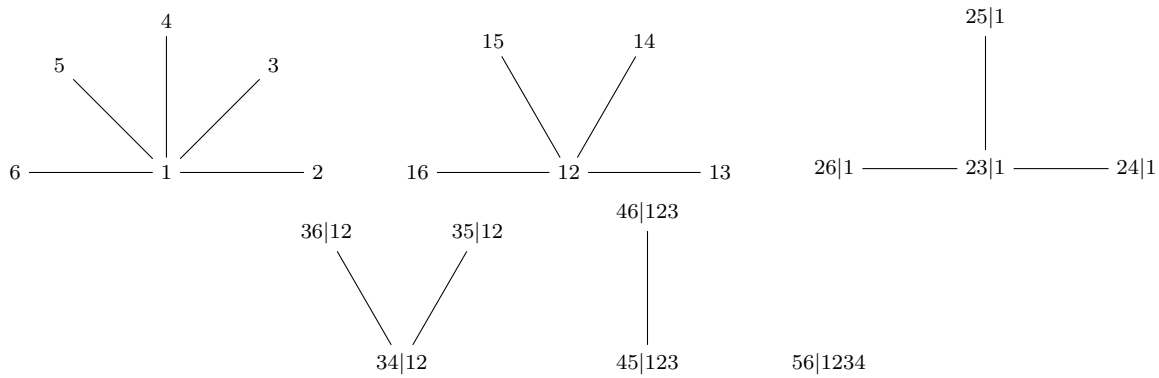


Fig. 2: Graphical representation of a C-vine with six nodes. The different trees graphically represent all bivariate copula functions needed to reconstruct the six dimensional joint density.

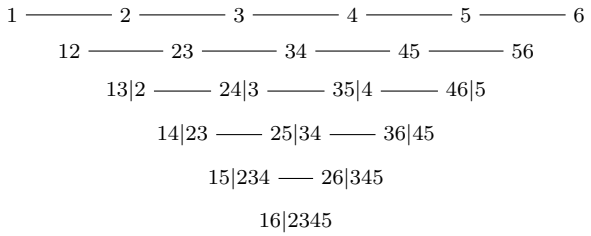


Fig. 3: Graphical representation of a D-vine with six nodes. The different trees graphically represent all bivariate copula functions needed to reconstruct the six dimensional joint density.

copulas for higher levels are obtained following the same reasoning. Once all copulas are specified for all levels, one reconstructs the joint density by multiplying all the copula functions as in (2.7).

While the trees and chains underlying C- and D-vines serve the purpose of creating a simple structure on pairs of nodes from which a multivariate copula can be constructed, the assumed connections between the random variables might not reflect true connections. For example, a C-vine relies on specifying one of the nodes as being a central node and then connects in a tree all other nodes to this one node. Also the 'chain' structure in a D-vine is quite specific and relies on an ordering of the variables, which may or may not make sense in practical applications. Our idea is to use the beneficial aspects of C- and D-vines, but to combine them with a DAG structure where a child node takes a central node and gets connected to its set of parents only (not necessary to all other nodes), with no particular order between the parental nodes.

3 Marrying DAGs and C/D-vines

We first construct a relation between a univariate conditional density and a C/D-vine representation. Next, using basic properties of copula density functions we show that a general multivariate density function can be decomposed into a product of vine ratios, where each vine can be further expressed as a product of bivariate conditional copulas. The DAG structure of the network indicates that the C/D-vines should be placed on particular nodes that are involved in moralized subgraphs in which the child node is a central node. Corollary 1 determines a precise structure which allows a rapid identification of the involved conditional copulas.

Without loss of generality assume we concentrate on a particular node, say x_l in the graph G. Conditional densities of the form $f_{l|pa(l)}(x_l|pa(x_l))$, where x_l is the child node and $pa(x_l)$ denotes the set of parents of node x_l, are the backbone of the entire factorization process involved in modeling data

using Bayesian networks. We assume that if the set of parents is the empty set then the conditional density is the same as the marginal density $f_l(x_l)$, and if $|pa(x_l)| = 1$ ($|S|$ denotes throughout the paper the cardinality of the set S), say $pa(x_l) = \{x_i\}$ then the definition of a copula density implies that the conditional density can be modeled as $f_{l|i}(x_l|x_i) = c_{l,i}\{F_l(x_l), F_i(x_i)\}f_l(x_l)$. For the case where $|pa(x_l)| = 2$, say $pa(x_l) = \{x_i, x_j\}$ the density $f_{l|i,j}(x_l|x_i, x_j)$ can be rewritten (non-uniquely) following basic definitions as $f_{l|i,j}(x_l|x_i, x_j) = c_{l,j|i}\{F_{l|i}(x_l), F_{j|i}(x_i)\}c_{i,j}\{F_i(x_i)F_j(x_j)\}f_l(x_l)$.

The above situations are to be contrasted with cases where $|pa(x_l)| > 2$ for which Lemma 1 specifies a particular ratio-based decomposition. For such cases, consider a general conditional density $f_{l|pa(l)}(x_l|pa(x_l))$ with $pa(x_l) = \{pa_1(x_l), pa_2(x_l), \ldots, pa_d(x_l)\}$ and define the extended set $l \cup pa(l) = \{x_l, pa_1(x_l), pa_2(x_l), \ldots, pa_d(x_l)\}$ containing $d + 1$ elements. We further associate to this set, a corresponding *index* set ${}^\star l \cup pa(l)$. For example, if a variable X_3 is conditioned on variables X_2, X_5, X_7, then the set $l \cup pa(l)$ in this particular case contains the variables $\{X_3, X_2, X_5, X_7\}$ with corresponding renamed index set ${}^\star l \cup pa(l) = \{1, 2, 3, 4\}$ having cardinality $|{}^\star l \cup pa(l)| = 4$. Thus the set $\{{}^\star X_1, {}^\star X_2, {}^\star X_3, {}^\star X_4\}$ corresponds to the set $\{X_3, X_2, X_5, X_7\}$. The purpose of this notation is to avoid ambiguity when writing the C/D-vine decompositions.

Lemma 1 *For a general conditional density $f_{l|pa(l)}(x_l|pa(x_l))$ with $|pa(x_l)| > 2$ there exist a C-vine and a D-vine representation such that*

$$f_{l|pa(l)}(x_l|pa(x_l)) = \frac{CV_l}{DV_l} f_l(x_l), \tag{3.1}$$

where

$$CV_l = \prod_{j=1}^{|{}^\star l \cup pa(l)|-1} \prod_{i=1}^{|{}^\star l \cup pa(l)|-j} c_{j,j+i|1,\ldots,j-1}\{F_{j|1,\ldots,j-1}({}^\star x_j|{}^\star x_1, \ldots, {}^\star x_{j-1}), F_{j+i|1,\ldots,j-1}({}^\star x_{j+i}|{}^\star x_1, \ldots, {}^\star x_{j-1})\},$$

$$DV_l = \prod_{j=1}^{|{}^\star l \cup pa(l)|-1} \prod_{i=2}^{|{}^\star l \cup pa(l)|-j} c_{i,i+j|i+1\ldots i+j-1}\{F_{i|i+1,\ldots,i+j-1}({}^\star x_i|{}^\star x_{i+1}, \ldots, {}^\star x_{i+j-1}), F_{i+j|i+1,\ldots,i+j-1}({}^\star x_{i+j}|{}^\star x_{i+1}, \ldots, {}^\star x_{i+j-1})\},$$

and ${}^\star x_j$ denotes the j-th random variable in the set $l \cup pa(l)$. The reverse is also true: starting from all bivariate copulas involved in CV_l and DV_l and all marginal distributions f_l, eq. (3.1) defines a valid conditional density function.

Proof From the definition of vines any joint density can be rewritten as *either* a C-vine or a D-vine. For this reason, without loss of generality assume a C-vine factorization for $f(x_l, pa(x_l))$ and a D-vine factorization for $f(pa(x_l))$. After simplifications the above expression follows. The reverse follows by using that both a C-vine and a D-vine decomposition determine a joint density function. So, we choose a C-vine that determines the joint density $f(x_l, pa(x_l))$ and a D-vine that determines $f(pa(x_l))$. The ratio between the two densities is the conditional density $f_{l|pa(l)}(x_l|pa(x_l))$.

The requirement in Lemma 1 of having more than two parents is needed for the D-vine decomposition. Although it seems notationally involved, Lemma 1 actually specifies at the level of the index set (rather than using the labels of the variables) where one needs to place bivariate copulas in a C- and D-vine construction in order to obtain a valid conditional density function. For example, the first factor in the C-vine decomposition (obtained when $j = i = 1$) is the unconditional copula $c_{1,2}\{F_1({}^\star x_1), F_2({}^\star x_2)\}$. Continuing with the example preceding Lemma 1, a bivariate copula function involving variables X_3 and X_2 is used. The '1' and '2' refer thus to the positions of the variables included in the set $l \cup pa(l)$. For example, to investigate the conditional distribution $f(x_3|x_2, x_5, x_7)$ we construct the corresponding sets $pa(l) = \{X_2, X_5, X_7\}$, where $l = 3$, $l \cup pa(l) = \{X_3, X_2, X_5, X_7\}$, ${}^\star l \cup pa(l) = \{1, 2, 3, 4\}$ and as a result the set $\{{}^\star X_1, {}^\star X_2, {}^\star X_3, {}^\star X_4\}$ will correspond in our approach to the set of variables $\{X_3, X_2, X_5, X_7\}$.

Inspecting Figures 2 and 3, our approach seems natural and is in line with the relations assumed by the researcher as we further elaborate. In a C-vine, at the first level, one node plays a central role,

in the sense of it being connected to all other nodes. Referring to any Bayesian network most of the knowledge discovery process concentrates on grasping which are the parents that influence the child node, or equivalently we concentrate first on 'incoming' edges at a child node. Looking at the C-vine, we see that its construction serves a similar purpose, since it links in the first tree a child (the central node) to all of its parents. While any node can be central, if one accepts the child as a central node then a nice concordance emerges between the vine and the subgraph of the Bayesian network where we focus on the child. Thus using the child as the central node has the advantage that all the incoming edges to a child (and only those) in the DAG are represented by undirected edges in the first tree of a C-vine. Moreover, since the remaining variables play the role of parents, it seems natural to give them equal importance, and a D-vine representation fits well for this purpose. In order to get a conditional density $f(\text{child}|\text{parents})$ we specify the joint density $f(\text{child, parents})$ by using a C-vine decomposition. For the density of the parents, one could use a C-vine decomposition too, but conceptually this puts one parent as a root parent linking it with all others, which contradicts the aim of treating parents as 'equally' important. The D-vine decomposition more closely reflects such an interest.

The key point in our method, is that the root node is selected according to a DAG structure. Czado et al (2012) guide this choice by inspecting bivariate association measures and choosing the one that maximizes it, as an alternative to arbitrarily placing a root node based on preferences.

Lemma 2 motives why we represent a ratio decomposable copula density function with the help of BNs. If $f(x_1,\dots,x_p)$ decomposes according to G, then also $c(x_1,\dots,x_p)$ decomposes according to the graph. This makes the first link between modeling Bayesian networks and copulas by specifying that a general multivariate copula density function can be decomposed based on the nodes in the DAG. Going from the joint density to a DAG is not a one-to-one process, as two or more DAGs might be Markov equivalent (which is to say they represent the 'same' list of conditional independencies) and thus represent the same joint density. This holds too when switching from a joint density function to a joint copula density.

Lemma 2 *Let G be a DAG with nodes corresponding to each variable from the random vector $\boldsymbol{X} = (X_1,\dots,X_p)$ and let the joint density $f(x_1,\dots,x_p)$ be decomposable as $f(x_1,\dots,x_p) = c(F_1(x_1),\dots,F_p(x_p))\prod_{l=1}^{p} f_l(x_l)$. If $f(x_1,\dots,x_p)$ decomposes according to G then, under the conditions of Lemma 1, the joint copula density also decomposes according to G,*

$$c(F_1(x_1),\dots,F_p(x_p)) = \prod_l \frac{CV_l}{DV_l}.$$

Proof From (2.2) we have that $f(x_1,\dots,x_p) = \prod_{l=1}^{p} f(x_l|pa(x_l))$ and based on Lemma 1 we have the identity $f_{l|pa(l)}(x_l|pa(x_l)) = \frac{CV_l}{DV_l} f_l(x_l)$, which, put together, leads to

$$f(x_1,\dots,x_p) = \prod_{l=1}^{p} \frac{CV_l}{DV_l} \prod_{l=1}^{p} f_l(x_l).$$

Since it is assumed that one can rewrite the joint density as in (2.3),

$$f(x_1,\dots,x_p) = c(F_1(x_1),\dots,F_p(x_p)) \prod_{l=1}^{p} f_l(x_l)$$

we have after simplification the equality $c(F_1(x_1),\dots,F_p(x_p)) = \prod_{l=1}^{p} \frac{CV_l}{DV_l}$.

Of interest is also the reverse. Given one starts from a graphical structure and associates vine ratios at each node in the graph based on the graph structure, can we then recover a multivariate joint density? Lemma 3 can be thought of as a reverse of Lemma 2. It shows that associating the ratio $\{\frac{CV_l}{DV_l}\}$ at each node l in the graph G, leads to a valid joint density.

Lemma 3 *Let G be a DAG with nodes corresponding to each variable from the random vector $\boldsymbol{X} = (X_1,\dots,X_p)$ and let the set of functions $\{\frac{CV_l}{DV_l};\ l = 1,\dots,p\}$ be associated with the nodes of G. Under the conditions of*

Lemma 1, if all independencies that can be read from G are the same as those that are present in the joint density f (that is we do not make from the graph any conditional independence assumption that is not consistent with the joint density), the function

$$\prod_{l=1}^{p} \frac{CV_l}{DV_l} \prod_{l=1}^{p} f_l(x_l) = f(x_1, \ldots, x_p)$$

determines a valid density.

Proof Starting from the DAG G one has information about the structure of the parental set. Based on the parental set, associated with the nodes in the graph G and based on a fixed copula density c, one can construct the corresponding nodewise ratios $\{\frac{CV_l}{DV_l};\ l = 1, \ldots, p\}$.

According to Lemma 1 the following holds $\frac{CV_l}{DV_l} f_l(x_l) = f_{l|pa(l)}(x_l|pa(x_l))$ and thus

$$\prod_{l=1}^{p} \frac{CV_l}{DV_l} f_l(x_l) = \prod_{l=1}^{p} f_{l|pa(l)}(x_l|pa(x_l)) = f(x_1, \ldots, x_p),$$

where the last equality follows from the decomposition of the joint density according to graph G.

It is a well-known property (Lauritzen, 1996) that if f admits a factorization according to G then $A \perp B|C$ if the set of nodes A and the set of nodes B are separated by the nodes in set C in the graph $G^m_{an(A\cup B\cup C)}$, which is a moralized graph containing the ancestral set of $A \cup B \cup C$. More intuitively this means that in this moralized graph all paths from any element in A to any element in B must pass through a node in the set C, so if one were to remove the set C and all the links connecting its elements to elements from A and B, then all elements from the set A are completely separated from the ones in B. By moralization we mean the process where 'unmarried' parents having a common child get connected (or 'married') by an undirected link and all arrows get dropped. The moralized graph is thus an undirected graph. Moreover, if the joint density function $f(x_1, \ldots, x_p)$ admits a factorization according to G then the density factorizes also according to the moralized graph G^m and obeys the global Markov property relative to G^m (see Lauritzen, 1996, Lemma 3.21).

Let G_l be the subgraph obtained after eliminating all nodes that are not parents of node l in G, and let G^m_l be the moralized graph obtained after marrying all unmarried parents in G_l that have a child, and disregarding directionalities. Let G^{m*}_l be the subgraph obtained from eliminating all undirected links that connect to node l in the graph G^m_l. Figure 4 depicts such graphs.

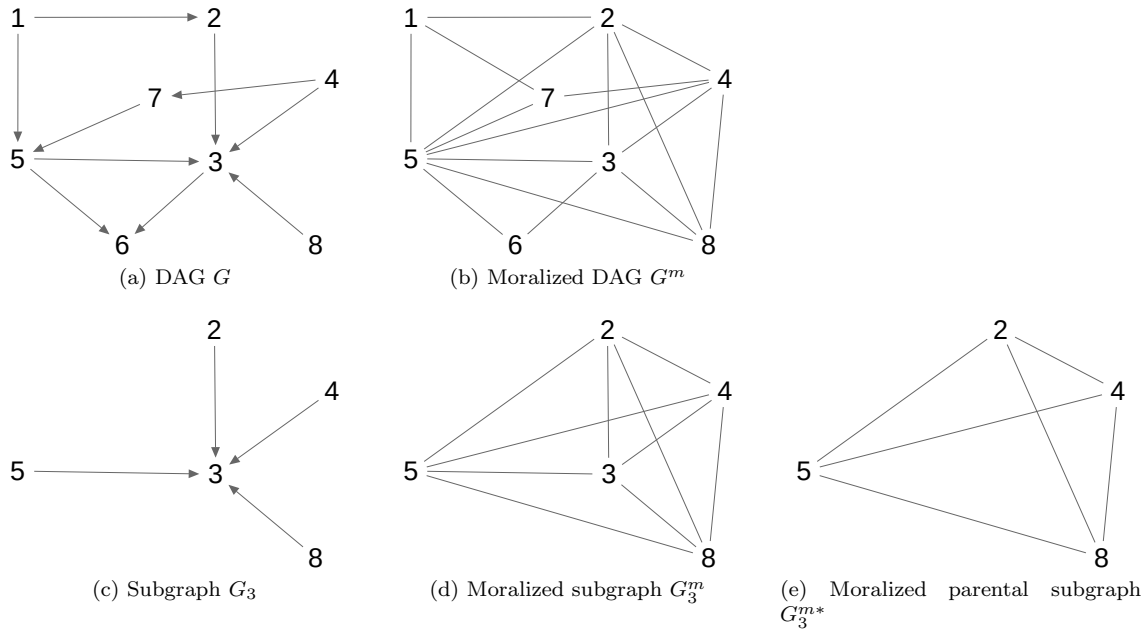
Theorem 1 makes the connection between copulas and Bayesian networks even more explicit. The main idea of our approach is the following: since the joint density factorizes according to a DAG and as well according to its moralized version, Theorem 1 specifies that in order to model data using bivariate copulas, one places a bivariate copula on each pair of variables that is connected by an edge in the moralized subgraphs. Thus the moralized graphs contain the key information about which variables need to be modeled with bivariate copulas in order to have a valid decomposition.

Theorem 1 *Let G be a DAG with nodes corresponding to each variable from the random vector $\boldsymbol{X} = (X_1, \ldots, X_p)$ and let the joint density $f(x_1, \ldots, x_p)$ decompose according to G. Under the conditions of Lemma 1, the joint density can be factorized as $f(x_1, \ldots, x_p) =$*

$$\prod_{l=1}^{p} \frac{f_l(x_l) \prod_{j=1}^{|^\star l\cup pa(l)|-1} \prod_{i=1}^{|^\star l\cup pa(l)|-j} c_{j,j+i|1,\ldots,j-1}\{F(^\star x_j|^\star x_1, \ldots, ^\star x_{j-1}), F(^\star x_{j+i}|^\star x_1, \ldots, ^\star x_{j-1})\}}{\prod_{j=1}^{|^\star l\cup pa(l)|-1} \prod_{i=2}^{|^\star l\cup pa(l)|-j} c_{i,i+j|i+1,\ldots,i+j-1}\{F(^\star x_i|^\star x_{i+1}, \ldots, ^\star x_{i+j-1}), F(^\star x_{i+j}|^\star x_{i+1}, \ldots, ^\star x_{i+j-1})\}}$$

where

(i) each bivariate copula $c_{j,j+i|1,\ldots,j-1}$ is set on each edge in G^m_l and,
(ii) each bivariate copula $c_{i,i+j|i+1,\ldots,i+j-1}$ is set on each edge in G^{m*}_l.



(a) DAG G (b) Moralized DAG G^m

(c) Subgraph G_3 (d) Moralized subgraph G_3^m (e) Moralized parental subgraph G_3^{m*}

Fig. 4: (a) DAG, (c) subgraph of the DAG, and moralized graphs (b, d and e).

Proof The expression follows from the application of (2.2) and Lemma 1. The joint density can be factorized as

$$f(x_1,\ldots,x_p)=\prod_{l=1}^{p} f_{l|pa(l)}(x_l|pa(x_l))=\prod_{l=1}^{p}\frac{CV_l}{DV_l}f_l(x_l)$$

and replacing CV_l and DV_l by their corresponding products of bivariate copulas, yields the density decomposition offered in the theorem.

For the decomposition

$$f(x_l,pa_1(x_l),\ldots,pa_d(x_l))=f(x_l|pa_1(x_l),\ldots,pa_d(x_l))\prod_{i=1}^{d}f_i(pa_i(x_l)),$$

the subgraph G_l corresponds to the conditional density $f(x_l|pa_1(x_l),\ldots,pa_d(x_l))$. We need to show that the moralized subgraph G_l^m can be used to represent the joint density function $f(x_l,pa_1(x_l),\ldots,pa_d(x_l))$ through bivariate copulas. We start by noting that G_l^m is a complete subgraph for which any pair of different nodes is linked and this implies that the graph contains $(d+1)d/2$ distinct edges. This is the same as the number of pairwise copulas used by a C-vine decomposition involving $d+1$ variables (d parents and 1 child node) where a copula function (conditional or unconditional) is placed on any two different nodes i and j. Both the C-vine and the graph G_l^m are constructed on the same nodes. Thus starting from the child node as the root node, one can construct the C-vine numerator by using $(d+1)d/2$ copulas where the pair of nodes $(j,j+1)$ is necessarily linked by an edge in G_l^m, because due to the completeness of the subgraph, to all different pairs $(j,j+1)$ there exists an edge that links them in the moralized graph G_l^m.

The second claim is obtained analogously. A D-vine is now used to model the parents and this gives rise to $d(d-1)/2$ different bivariate copulas involving two different nodes, the graph G_l^{m*} is again a fully connected graph, and the same reasoning as above applies. The simpler structure of G_l^{m*} as compared to G_l^m indicates that only in a C-vine the child-parent dependence is modeled.

The proof continues by induction. Assume the above holds for an arbitrary number of nodes n. Then having $n+1$ nodes does nothing else than enlarge the moralized subgraph with n extra edges, one connecting it to all other nodes and the number of bivariate copulas increases by the same amount.

T_1^C	1	2	3	4	5	6
T_2^C	12	13	14	15	16	
T_3^C	23|1	24|1	25|1	26|1		
T_4^C	34|12	35|12	36|12			
T_5^C	45|123	46|123				
T_6^C	56|1234					

T_1^D	2	3	4	5	6
T_2^D	23	34	45	56	
T_3^D	24|3	35|4	46|5		
T_4^D	25|34	36|45			
T_5^D	26|345				

Fig. 5: Tree-by-tree description of copulas involved in the numerator (left panel) and denominator (right panel), see Corollary 1.

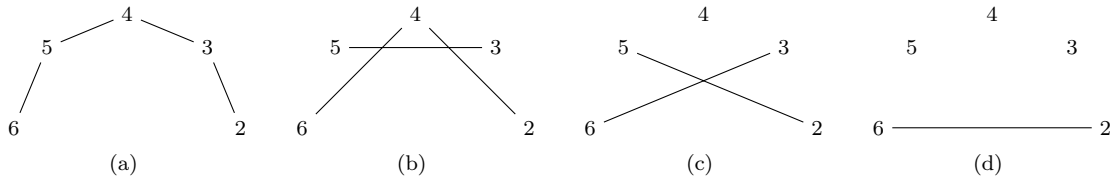


Fig. 6: Graphical representation of a D-vine with 5 nodes. Undirected edges couple nodes in bivariate copulas for different trees.

Our approach is different from Elidan (2010, 2012), since there higher dimensional copulas are used to model variables in the set $\{l \cup pa(l)\}$.

Corollary 1 shows that there is a systematic way in which pairwise copulas are introduced. We denote by T_r^C and T_r^D the r-th tree from the C-vine or D-vine and by $T_{r,s}^C$, resp. $T_{r,s}^D$ we denote the s-th node from the r-th tree in a C-vine, respectively D-vine.

Corollary 1 *Let $f_{l|pa(l)}(x_l|pa(x_l)) = \frac{CV_l}{DV_l}$ be the conditional density of variable X_l with the conditioning set of parents $pa(x_l)$ following the condition stipulated in Lemma 1. By construction, the nodes $T_{r,s}^C$ and $T_{s+1,r-2}^D$ involved in $\frac{CV_l}{DV_l}$, where $r = 3, \ldots, |pa(l)| + 1$ and $s = 1, \ldots, |pa(l)| + 2 - r$ require the bivariate copulas $c_{i,j|\{h:<\min(i,j)\}}$ in the case of the C-vine and $c_{i,j|\{h:\min(i,j)<h<\max(i,j)\}}$ in the case of the D-vine, with $i, j, h \in {}^{\star}l \cup pa(l)$.*

The example in Figure 5 shows a tree-by-tree description of the copulas to help appreciate the corollary. For $r = s = 3$ we have in the third tree, as the third node the bivariate copula $c_{2,5|1}$ in the C-vine decomposition. Based on the corollary, in the D-vine decomposition in the fourth tree, as the first element, we have a conditional copula on the *same* variables, the parents, but using the set $\{3, 4\}$ as conditioning set. All copulas involved in both panels can be recovered from Figures 2 and 3 where we also explained how the vine graphs can be read.

Corollary 1 clearly expresses which variables should be involved in the numerator and the denominator and how moving from one tree to another impacts the respective conditioning. Keeping the same order for the parents in both the C-vine and the D-vine results in the compact representation of Corollary 1, see also Figure 5.

The same information from the right hand side of Figure 5, can be graphically expressed by starting from the first tree of a C-vine where we retain only the leafs that act as parents. Figure 6 lists graphically all the nodes that get connected and all the pairwise connections in the D-vine. On each edge a bivariate copula is set, and the conditioning set is represented by all nodes in between. For example, in the third tree (panel b) one has the conditional copula $c_{2,4|3}$ (alongside $c_{3,5|4}$ and $c_{4,6|5}$), in the fourth tree (panel c) the conditional copula $c_{2,5|3,4}$ (alongside $c_{3,6|4,5}$) is being used and in the fifth tree (panel d) the conditional copula $c_{2,6|3,4,5}$ is used.

4 Motivating example

We now return to the 'Wine' data and present a practical application of the cDAGs method. A rigorous explanation of the computational aspects is postponed to Sections 5 and 6. We restrict here to applying the proposed procedure.

Figure 7 presents two estimated cDAGs for the Wine data, for which the structure is estimated as described in Section 5. For illustration we omit the variable 'Hue' when estimating the cDAG in panel (a). In both cases the Gaussian, Clayton, Gumbel, Frank and Joe copula families have been used as candidate families in the cDAG procedure.

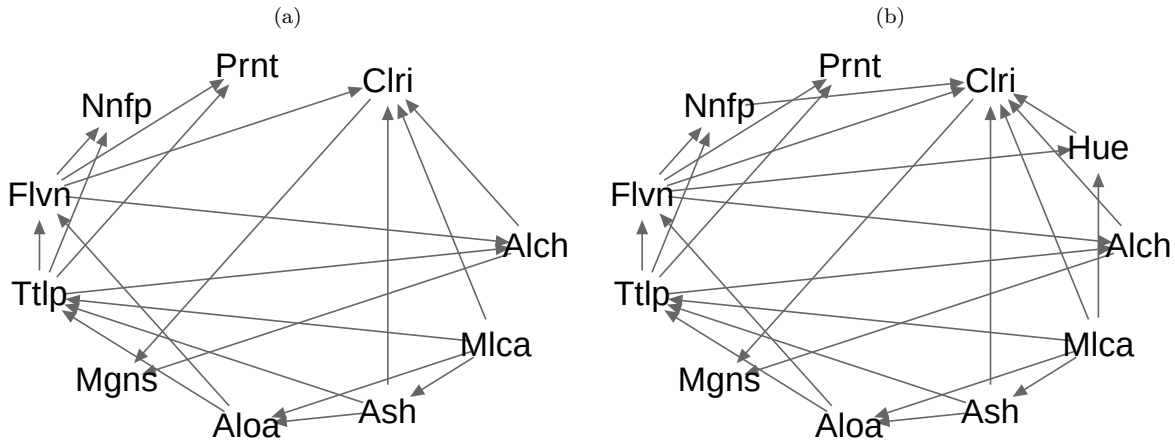


Fig. 7: Wine data. Estimated cDAGs using either 10 (a) or 11 (b) variables.

Compared to the PC and BGe graphs presented in Figure 1(a),(b) in Section 1 it is apparent that the cDAG is slightly denser and estimates more edges than the other two techniques for this dataset. The main difference is that in Figure 7 a series of bivariate copulas has been used instead of a multivariate normal distribution. Compared to the C-vine in Figure 1(c) the new approach places bivariate copulas on nodes that are in a sense 'causally' connected whereas the C-vine decomposition is more rigid and given a central node, all others get connected to it, although modeling such a dependence structure might not be warranted in the multivariate distribution. On the other hand a BN tries to capture exactly the way in which variables influence each other.

The benefit of incorporating the structure of the DAG for deciding which two nodes need to be coupled by the copula is observed in a higher log-likelihood as compared to a C-vine as implemented in Brechmann and Schepsmeier (2013), for which we have used the BIC criterion to obtain the final model and for selecting the best fitting copulas from the same list of copula families as for the cDAG. For the 10 variables case the log-likelihood values are 477 for the cDAG and 420 for the C-vine and for the 11 variables case the log-likelihood values are 557 for the cDAG and 494 for the C-vine. Interesting to note is that for the 10 dimensional data the cDAG method achieves a higher log-likelihood with less bivariate copulas than the C-vine method, while in the 11 dimensional case, cDAG needs more copula terms than the C-vine. Moreover, an out-of-sample prediction evaluation, described in Section 7, on two measures of dependence, namely Kendall's τ and Gini's index, the cDAG outperforms the C-vine by incorporating the structure of the data.

Regarding structure resemblance, we have compared the obtained cDAG structures with the following DAG finding algorithms: hill-climbing (HC) based on AIC/BIC or BGe, the PC algorithm as described in Kalisch and Bühlmann (2007) and the SIN algorithm (Drton and Perlman, 2008). We emphasize that all these algorithms make the explicit assumption of multivariate normality. We expect some overlap in the identified edges by cDAG and these procedures, but not to a very high degree, because of the different distributional assumptions. See Section 5 for details regarding the model selection criterion we use. Table 1 shows for both data examples an overlap between the estimated graphs. For

this dataset, the cDAG method estimates graphs which are closer to the hill-climbing based on the AIC and further away from the SIN estimated graphs using the cut-off value of $\alpha = .1$.

	10 variables	11 variables
HC-BGe	40%	42%
HC-BIC	38%	40%
HC-AIC	42%	42%
PC(α =.1)	40%	36%
PC(α =.05)	38%	35%
SIN(α =.1)	31%	31%
SIN(α =.05)	33%	31%

Table 1: Wine data. The graphs presented in Figure 7 are compared to DAGs estimated with popular algorithms with respect to the proportion of common edges in the skeleton of the estimated DAGs. The dataset contains either 10 or 11 variables.

5 Model selection in parametric families

We start by presenting a parametric estimation framework and proceed by studying an information criterion for simultaneously selecting the copula family and the structure of the DAG.

5.1 Parametric copula families

The cDAG method is developed in Section 3 at the probabilistic level. From a statistical viewpoint we estimate such structures along with corresponding parameters from samples, as has been done for the example in Section 4.

Given n realizations of a p-dimensional random vector $\boldsymbol{X}_k = (X_{k1}, \ldots, X_{kp})$ with $k = 1, \ldots, n$ we construct the pseudo-observations, by first retaining the rank r_{ki} of the variable X_{ki} among all variables $\{X_{1i}, \ldots, X_{ni}\}$ and then scale it by a factor $n+1$ to ensure that all values are inside $(0, 1)$,

$$\widetilde{X}_{ki} = \frac{r_{ki}}{n+1} = \frac{\sum_{t=1}^{n} \mathbf{1}(X_{ti} \leq X_{ki})}{n+1} = \frac{n}{n+1} F_{n,i}(X_{ki}),$$

with $F_{n,i}$ the empirical distribution function of the ith component of the p-vector. We will use everywhere in our calculations the pseudo-observations instead of the real observations. The pseudo-observations have marginal distributions that are approximately uniform on $[0, 1]$ and hence we set the values of the densities equal to 1.

All the bivariate copulas $c_{i,j|k,l}(\cdot, \cdot; \boldsymbol{\theta}_{i,j|k,l})$ are modeled with an unknown generally low-dimensional vector of parameters $\boldsymbol{\theta}_{i,j|k,l}$ that can be estimated from the data. In this paper we have used the following parametric copulas (other choices are possible): the Gaussian copula with $C_\theta(u, v) = \Phi_\theta(\Phi^{-1}(u), \Phi^{-1}(v))$, the Clayton family with $C_\theta(u, v) = \{\max(u^{-\theta} + v^{-\theta} - 1,\ 0)\}^{-1/\theta}$, the Gumbel family with $C_\theta(u, v) = \exp[-\{(-\log u)^\theta + (-\log v)^\theta\}^{1/\theta}]$, the Frank family with $C_\theta(u, v) = -\frac{1}{\theta}\log(1 + \frac{\{\exp(-\theta u)-1\}\{\exp(-\theta v)-1\}}{\exp(-\theta)-1})$ and the Joe family with $C_\theta(u, v) = 1 - \{(1-u)^\theta + (1-v)^\theta - (1-u)^\theta(1-v)^\theta\}^{1/\theta}$.

In a parametric framework, Theorem 1 gives a factorization of the joint density as

$$f(x_1, \ldots, x_p; \boldsymbol{\theta}) = \prod_{l=1}^{p} \frac{CV_l(\boldsymbol{\theta}_{CV_l})}{DV_l(\boldsymbol{\theta}_{DV_l})} \prod_{l=1}^{p} f_l(x_l),$$

where $\boldsymbol{\theta}_{CV_l}$ and $\boldsymbol{\theta}_{DV_l}$ are vectors of parameters resulting from the copula families and $\boldsymbol{\theta} = (\boldsymbol{\theta}_{CV_l}, \boldsymbol{\theta}_{DV_l})$ is the combined parameter vector.

Given n realizations of the random vector X_k, the log-likelihood of the data is $\ell(\boldsymbol{\theta}; X_1, \ldots, X_n) = \sum_{k=1}^{n}\sum_{l=1}^{p}(\log CV_{l,k} - \log DV_{l,k})$, where the subscript k in $CV_{l,k}$ and $DV_{l,k}$ indicates the use of the kth observation.

Note that the log-likelihood is not in general monotone in the number of parameters, since a 'bad' model for the C-vine might outweigh a 'good' model for the D-vine. This likelihood may be nodewise decomposed as $\ell(\boldsymbol{\theta}; X_1, \dots, X_n) = \sum_{l=1}^{p} \text{log-Lik}(\boldsymbol{\theta}; \text{node}_l)$.

Parameter estimation proceeds sequentially, see Czado et al (2012) and Hobæk Haff (2013). Using a tree-by-tree decomposition as in Figure 5, the estimates at a level q are found by plugging-in estimates obtained at the level $q-1$, and so on.

5.2 A nodewise information criterion

We define at each node l in the graph the nodewise penalized score

$$IC_l = -2\text{log-Lik}(\hat{\boldsymbol{\theta}}; \text{node}_l) + \widehat{\text{pen}}(n, \hat{\boldsymbol{\theta}}), \tag{5.1}$$

where the first part is the nodewise log-likelihood value of the model at the estimated parameter $\hat{\boldsymbol{\theta}}$ and the second part is a penalty expressing the complexity of the model as a function of $\hat{\boldsymbol{\theta}}$ and the sample size n. The smaller the value of the information criterion, the better the model. The penalty can take various forms and popular penalties have been proposed in the literature, the most famous ones are the Akaike information criterion (Akaike, 1973) AIC with $\widehat{\text{pen}}(n, \hat{\boldsymbol{\theta}}) = 2\,\text{length}(\hat{\boldsymbol{\theta}})$ and the Bayesian information criterion (Schwarz, 1978) BIC with $\widehat{\text{pen}}(n, \hat{\boldsymbol{\theta}}) = \text{length}(\hat{\boldsymbol{\theta}}) \log n$. In a nodewise information criterion the penalty is based on the parent configuration of the considered node. A value of the information criterion of the complete graph G is defined as the sum of the nodewise values, $IC(G) = \sum_{l=1}^{p} IC_l$.

We motivate the use of a new penalty which bridges two aspects: the DAG structure and the copula based representation of the nodewise conditional density. As in (5.1) we define a nodewise information criterion using $\widehat{\text{pen}}_{\text{cDAG}}(n, \hat{\boldsymbol{\theta}}) = 2\sum_{k=1}^{n} \log DV_{l,k} / \{|pa(x_l)| \log n\}$,

$$\text{cDAG-IC}_l = -2\sum_{k=1}^{n} (\log CV_{l,k} - \log DV_{l,k}) + \frac{2\sum_{k=1}^{n} \log DV_{l,k}}{|pa(x_l)| \log n},$$

where the log-likelihood expresses how the conditional density is represented by ratios of C/D-vines to model the child-to-parents and the parents-to-parents relation. The corresponding graph score is then $\text{cDAG-IC}(G) = \sum_{l=1}^{p} \text{cDAG-IC}_l$.

The penalty part uses the number of estimated parents of a node and the relative 'cost' per parent as opposed to the number of estimated parameters. Thus the larger the cost of modeling the parents the higher the reduction in the likelihood part and the larger the penalty. The reasoning behind it is that we are interested in nodewise models where the benefits of modeling the parents (which we have argued is of secondary importance, but nonetheless informative) should not come at the expense of modeling the child-parents relation. The proposed penalty satisfies consistency properties (see Section 5.3), though other definitions of $\widehat{\text{pen}}(n, \hat{\boldsymbol{\theta}})$ can be explored too. Note that by Lemma 4 we could replace the $\log n$ factor by any sequence a_n such that $a_n \to \infty$ and $a_n = o(\sqrt{n})$ as $n \to \infty$.

By estimating the structure from the data, the cDAG method might often be more complex than using a simple C- or D-vine. Hence only counting the number of parameters as in an information criterion such as AIC or BIC, is not appropriate (see Section 7).

The nodewise decomposability of both the models and the information criterion allows for fast parallel estimation using the algorithms of Aas et al (2009).

5.3 Consistency of nodewise model selection

Without loss of generality, consider two models to choose from at any given node l. The models can specify different copula families or can differ in the structure of the parental set. A superscript indicates the model used, e.g., CV_l^m, DV_l^m, $pa^m(l)$ denote the quantities CV_l, DV_l and $pa(l)$ for model $m \in \{1, 2\}$.

We adapt the general assumptions of Sin and White (1996, Propositions 4.1 & 4.2(a,b)) to our context, which for completeness are stated in the appendix.

Lemma 4 (Adapted from Proposition 4.2 of Sin and White (1996)) *Under the assumptions as stated in the appendix, for both models, let* $\Delta cDAG\text{-}IC_l = cDAG\text{-}IC_l^1 - cDAG\text{-}IC_l^2$, *and with the expectation computed with respect to the true density, let* $\Delta_n = n^{-1}(E[log\text{-}Lik^1(\boldsymbol{\theta}_{0n}^1; node_l) - log\text{-}Lik^2(\boldsymbol{\theta}_{0n}^2; node_l)])$ *with* $\boldsymbol{\theta}_{0n}^m$ *the least false parameter value (see condition iv in the Appendix) and* $\Delta\widehat{\text{pen}} = \widehat{\text{pen}}^1(n, \hat{\boldsymbol{\theta}}^1) - \widehat{\text{pen}}^2(n, \hat{\boldsymbol{\theta}}^2) > 0$.
(i) If $\liminf_{n\to\infty} \Delta_n > 0$ *and the penalty satisfies that* $\Delta\widehat{\text{pen}} = o_P(n)$, *then weak consistency holds, that is,* $\lim_{n\to\infty} P(\Delta cDAG\text{-}IC_l < 0) = 1$.
(ii) If $\limsup_{n\to\infty} n^{1/2}\Delta_n < \infty$, *for* $m = 1, 2$ *the terms in log-Lik*$^m(\boldsymbol{\theta}_{0n}^m; node_l)$ *satisfy a central limit theorem and* $P(n^{-1/2}\Delta\widehat{\text{pen}} \to \infty) = 1$, *then* $\lim_{n\to\infty} P(\Delta cDAG\text{-}IC_l \geq 0) = 1$.
(iii) If log-Lik$^1(\boldsymbol{\theta}_{0n}^1; node_l)$ − *log-Lik*$^2(\boldsymbol{\theta}_{0n}^2; node_l) = O_P(1)$ *and* $P(\Delta\widehat{\text{pen}} \to \infty) = 1$ *then it holds that* $\lim_{n\to\infty} P(\Delta cDAG\text{-}IC_l \geq 0) = 1$.

The consistency result for the graph information criterion holds by the nodewise decomposability of $IC(G)$. In the appendix it is shown that the conditions on $\Delta\widehat{\text{pen}}$ in (i)–(iii) of Lemma 4 (short, the penalty conditions) hold for cDAG-IC$_l$.

The penalty requirement in (i) is satisfied for AIC, BIC and also for the cDAG-IC. When there is one clear winner in terms of Kullback-Leibler distance to the true density of the data, all three criteria choose with probability going to one the Kullback-Leibler best model. For the situation of graphical modeling, a situation with $\liminf_{n\to\infty} \Delta_n > 0$ occurs for example when one graph (say G_2) is missing one or more true edges and the other graph G_1 includes the needed edges. In the limit, the information criteria are able to identify G_1 as the better graph.

The penalty conditions in (ii) and (iii) hold for the BIC and for cDAG-IC though not for AIC. As a consequence, when the models are close BIC and cDAG-IC select with probability going to one the model with the smallest penalty, often referred to as the most parsimonious model. For example, in the case of graphical models such a situation occurs when one graph contains one or more edges too many. If it truly holds that $X_i \perp X_j | pa(X_j)$ but graph G_1 still includes an edge between i and j and graph G_2 does not, both graphs are decomposing the same density (conditioning on an independent variable causes no harm), having $\Delta_n = 0$. Under the stronger assumption on the penalty in (ii), then the more parsimoneous graph G_2 is correctly identified with probability going to one. For some consistency results for selecting graphs, see also Chickering (2002).

In a similar way we can show that, provided the conditions for strong consistency as in Sin and White (1996, Proposition 5.2(a)) hold, the proposed penalty in cDAG-IC satisfies that $\Delta\widehat{\text{pen}} = o(n)$ almost surely, and thus strong consistency may be obtained.

5.4 Across-class model selection

The cDAG-IC has been constructed to select and compare models *within* the class of cDAG models. Selecting from distinct non-nested models (e.g., a DAG, a cDAG, a C-vine) requires additional care. Traditional AIC and BIC can be used for this purpose since they compare the overall fit of the graph and penalize it with a function of the number of parameters estimated by the model. For a small number of comparisons one may perform alternatively hypothesis testing as in Vuong (1989) and Clarke (2003). In Section 8 we apply such an across-classes selection using AIC, BIC and Vuong's likelihood ratio hypothesis tests.

6 Computational aspects

We implemented the cDAG procedure using the statistical software R (R Core Team, 2014). We start by specifying a set of copula density families. At each node l, the algorithm computes the cDAG-IC score based on the parental set at that step of the algorithm (see below) and for each copula family in the candidate set. We retain the copula that has the smallest cDAG-IC value. For each node we simultaneously select a copula family used per ratio $\frac{CV_l}{DV_l}$ and a parental set. The final result is an estimated DAG and a set of copula families which are used in the ratio decomposition at each node.

For computational simplicity once a copula family is selected at node l then both the numerator and denominator in $\frac{CV_l}{DV_l}$ use it for all involved bivariate copulas. However, at different nodes different

Algorithm 1 Information criterion based search method for cDAGs

$\hat{G} \leftarrow$ `empty graph`
cDAG-IC$(\hat{G}) \leftarrow \infty$
$Flag \leftarrow False$;
while $Flag == False$ **do**
  `compute` Add `based on` $\hat{G}$;
  `compute` $Delete$ `based on` $\hat{G}$;
  `compute` $Invert$ `based on` $\hat{G}$;
  $Allmoves \leftarrow$ `append` $Add, Delete, Invert$;
  $Length \leftarrow$ `the length of` $Allmoves$;
  for $\hat{G}^{current} \in Allmoves$ **do**
    `compute` cDAG-IC$(\hat{G}^{current})$ `score as follows:`
    for $l \in$ `updated nodes` **do**
      cDAG-IC$_l \leftarrow -\infty$
      for $copula \in$ `Used copula family set` **do**
        cDAG-IC$_l^{copula} \leftarrow$ `compute` cDAG-IC$_l$ `using copula`
        if `minimum`(cDAG-IC$_l^{copula}$) $<$ cDAG-IC$_l$ **then**
          cDAG-IC$_l \leftarrow$ cDAG-IC$_l^{copula}$
          `Selected Copula` $\leftarrow copula$
        end if
      end for
      cDAG-IC$(\hat{G}^{current}) \leftarrow \sum_{l=1}^{p}$ cDAG-IC$_l$
    end for
  end for
  if `minimum`(cDAG-IC$(\hat{G}^{current})$) $<$ cDAG-IC$(\hat{G})$ **then**
    $position \leftarrow$ `position of minimum`(cDAG-IC$(\hat{G}^{current})$);
    $\hat{G} \leftarrow \hat{G}^{current}[position]$;
    cDAG-IC$(\hat{G}) \leftarrow$ `minimum`(cDAG-IC$(\hat{G}^{current})$);
  else
    $Flag \leftarrow True$;
  end if
end while

copula families can be selected. Our methodology generalizes directly to cases where one desires to select different copula families at the level of the numerator or denominator, but this is much more computer intensive and for large dimensional problems it can be quite cumbersome.

The presented procedure is based on a 'divide and conquer' approach, where a relatively complex and 'hard-to-solve' problem is split into several smaller manageable problems which can be quickly solved using nodewise modeling.

Algorithm 1 contains pseudo-code for the implementation. The algorithm starts from an empty graph and updates its structure according to a hill-climbing procedure, if the updated graph improves the current value of the information criterion. In a hill-climbing procedure three steps are allowed: add a directed edge between two nodes, remove a directed edge or invert the directionality of an edge. The DAG is updated and modified according to the smallest value of the information criterion.The algorithm ends when the value of the information criterion cannot be improved.

This procedure is a local optimization technique in the space of DAGs, which avoids the (sometimes impossible) listing of all graphs. Since at each step at most two nodes change their parental sets, only at most two values of the nodewise information criterion need to be updated. To make the nodewise procedure more clear, we present in Algorithm 2 pseudo-code for the followed steps at each node. For any given graph we analyze the nodes separately. Based on the implied parents of the current node, C-vines and D-vines decompositions are obtained. Next, based on the estimated parameters we construct the nodewise contributions to the total graph score cDAG-IC. We retain incrementally the best fitting copula family as well as the best fitting set of parents. Note that in the search algorithm it is quite possible for the best fitting copula family to change as every time different sets of parents are used.

Algorithm 2 Nodewise procedure

1. For each node l in a given graph G based on directed edges, extract the set $pa(l)$;
2. For each copula family in a specified list of families, estimate parameters for all bivariate copulas in $\frac{CV_l}{DV_l}$;
3.1. For CV_l, using the set $l \cup pa(l)$ perform the C-vine decomposition;
3.2. For DV_l, using the set $pa(l)$ perform the D-vine decomposition;
4. Using the estimated parameters calculate the value nodewise contribution to the total graph score, i.e, cDAG-IC$_l$
5. Select the copula family which minimizes the value cDAG-IC$_l$;
6. Accept directed arrow (or the set $pa(l)$) if total graph score has improved over the baseline

7 Simulations

To evaluate the performance of the cDAG method we have set-up in this section a simulation study under a variety of settings which differ in number of nodes, sample sizes, estimation method and ways of generating data used in the estimation proces.

7.1 Simulation settings A

We have generated data (i) from a DAG, using the 'pcalg' package (Kalisch et al, 2012) in R , where at each node we have added random errors that have either a Student t(df=4) distribution (denoted t4 throughout the section) or a standard Gaussian distribution; (ii) from a general multidimensional copula. The popular Clayton, Gumbel and Frank families have been used with $\theta \in \{3.3, 1.3, 0.5\}$ as parameters for the three copula families. Data have been generated using the 'copula' package (Hofert et al, 2014) in R. Using the same families we have also generated data from hierarchical Archimedean copulas using the 'HAC' package (Okhrin and Ristig, 2014), where the parameter vectors were drawn at random uniformly in the interval [2,6]. In case the data were generated from a DAG, the probability (π) of connecting two nodes was either 0.1 or 0.4, the lower the value, the less directed edges the DAG contains. The sample size was 200 or 1000 and the number of nodes p varied in the set $\{5, 10, 20, 25\}$.

For the cDAG method we have used the cDAG-IC to simultaneously select the final structure and the copula families. We also investigated the case of fixing the Gaussian copula and only select the structure of the DAG. For the C-vine approach we used the publicly available R package 'CDVine' (Brechmann and Schepsmeier, 2013) to select a copula for each term in the decomposition based on their standard BIC implementation, and considered also a second case where the Gaussian copula models all terms. We have also compared our approach against a more general R-vine, as implemented in the 'VineCopula' package (Schepsmeier et al, 2014).

For all techniques the list of copulas from which selection was desired was set to contain the Gaussian, Clayton, Gumbel, Frank and Joe copula families.

For the C- and R-vines we allowed also the independence copula to be in the list of copulas to be chosen in order to reduce the complexity of the vine models. The independence test offered in Genest and Favre (2007) was used. Since the test uses formal hypothesis testing which is applied to each bivariate term in the decomposition, for large dimensional problems it can result in accumulating type-I errors. We did not use any correction factors to account for the multiple testing problems arising when selecting the independence copula and just used the vines software as is offered.

We compare the cDAG method to the competitor procedures using four characteristics:

1. the estimated log-likelihood of the data under the specified model;
2. the number of parameters of the models;
3. $SSQ_\tau = \sum_{\text{all pairs (i,j)}} (\tau_{ij}^{obs} - \tau_{ij}^{sim})^2$: the sum of squared differences between the observed Kendall's τ computed on nodes i and j and a corresponding τ estimated on simulated data based on the model;
4. $SSQ_{\text{Gini}} = \sum_{\text{all pairs (i,j)}} (\text{Gini}_{ij}^{obs} - \text{Gini}_{ij}^{sim})^2$: the sum of squared differences between the observed Gini's index computed on nodes i and j and a corresponding Gini's index estimated on simulated data based on the model.

In characteristics 3–4 (but not in 1 and 2) a further level of simulation is used. The purpose of investigating these two characteristics is to show the potential benefits a local or nodewise structure delivers when

Algorithm 3 Simulation settings

Data coming from DAG
1. Set the number of nodes $p \leftarrow 5, 10, 20$ or 25 and sample size $n \leftarrow 200$ or 1000 cases;
2. Set the probability of connecting two nodes $\pi \leftarrow 0.1$ or 0.4;
3. Set random edge weights with values between $w \in [.05,\ .3]$;
4. Generate a DAG G with p nodes, probabilities π and weights w;
5. Based on G generate multivariate data (of size $n \times p$);
6. At each node add random noise coming from N(0,1) or Student-t (df=4) distribution;
7. Transform data to pseudo-observations to be used in all calculations;

Data coming from Clayton/Gumbel/Frank copulas;
1. Set the number of nodes $p \leftarrow 5, 10, 20$ or 25 and sample size $n \leftarrow 200$ or 1000 cases;
2. Set copula families to Clayton($\theta = 3.3$), Gumbel($\theta = 1.3$) and Frank ($\theta = 0.5$);
3. From each of the copulas generate multivariate data (of size $n \times p$) to be used further;

Data coming from hierarchical Clayton/Gumbel/Frank copulas;
1. Set the number of nodes $p \leftarrow 5, 10, 20$ or 25 and sample size $n \leftarrow 200$ or 1000 cases;
2. For each copula families generate a vector of parameters θ of length $p-1$ uniformly on in the interval [2,6];
3. From each of the copulas generate multivariate data (of size $n \times p$) to be used further;

Estimation and selection;
1. Set list of plausible copula families to {Gaussian, Clayton, Gumbel, Frank, Joe};
2. For cDAGs estimate the structure of the graph and the copula families for each ratio using the cDAG-IC criterion;
3. For C- and R-vines estimate the best fitting copula families using BIC and allow for independence screening;

Evaluation;
1. For all competitors investigate the estimated log-likelihood (ℓ) and the number of estimated parameters;
2. For each pair of nodes compute on the observed data Kendall's τ and Gini's index i.e., $\tau_{ij}^{obs}$and Gini_{ij}^{obs};
4. For each fitted model (using estimated parameters) simulate new data of size $n_2 \times p$, where $n_2 = 100$;
5. For each pair of nodes compute on the new simulated data Kendall's τ and Gini's index i.e., $\tau_{ij}^{sim}$and Gini_{ij}^{sim};
6. Compare all *obs* to *sim* quantities.

controlling for the influence of the copula. We wish to see experimentally if bringing information about the parental sets improves the accuracy of estimating these quantities, as small parental sets would be indicative that some of the extra terms involved in a regular C- or R-vine would be superfluous.

For every pair of nodes, on the observed dataset we compute Kendall's τ and Gini's index, and label them with the superscript *obs*. Next, a cDAG using the cDAG-IC criterion is constructed. Based on the estimated graph and its corresponding C-vine numerators and estimated parameters, we generate a new dataset for which we again compute for each pair of nodes i, j the τ value and Gini's index, this time denoted with the superscript *sim*. We stress that the new dataset is generated based on the cDAG proposed model in order to evaluate its performance.

We repeat the process of generating new data for the vine models as well. To evaluate the effect of the structure of the cDAG has on the estimation of the two quantities, we keep the copula effect under control, by using for both the C- and R-vine models the copula families selected by the cDAG. We then estimate the C- and R-vine models and based on the estimated parameters of the fitted vine models, we generate a new dataset. We then compute again for each pair of nodes i, j the τ value and Gini's index, which are labeled also with superscript *sim* to denote that they come from a generated dataset.

In the end, for each technique we take as a measure of performance the squared differences between the quantities estimated on the observed dataset and the quantities estimated on the generated datasets. Differences between quantities will thus be the effect of modeling the structure of the graph and will be unrelated to the choice of the copula whose effect is alleviated. In this way, if the estimated cDAG reflects an appropriate structure for the data, then generating data using the sets $l \cup pa(l)$ with $l = 1, \ldots, p$, is expected to produce on average samples which come closer to the original data, than when sampling from the global C- or R-vine where all nodes are involved in the data generation. Hence, smaller differences between such observed and simulated values indicate a better performance.

A total of 350 different observed datasets are generated and for each of them, we each time simulate three new datasets that come from the cDAG model and the C- and R-vine models for evaluating

			Estimated copula						Gaussian copula					
			log-likelihood			No. Parameters			log-likelihood			No. Parameters		
Data	p	n	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine
t4	5	200	10.7	5.4 *	5.5 *	14.3	1.1	1.1	8.6	8.5	8.5	15.8	10	10
t4	5	1000	25.6	20.6 *	20.5 *	13.9	1.3	1.3	23.3	23.2	23.2	15.4	10	10
t4	25	200	278.2	177.4 *	181.4 *	846.6	33.8	33.5	252.6	277.2	277.2	762.9	300	300
Gaussian	5	200	23.3	17.8 *	18.0 *	12.4	2.8	2.8	20.8	19.9	19.9	13.1	10	10
Gaussian	5	1000	84.8	77.8	77.7	10.7	4.1	4.0	82.6	78.9	78.9	11.2	10	10
Gaussian	25	200	879.1	767.9 *	754.4 *	245.1	92.4	79.8	842.8	861.9	861.9	147.5	300	300
Gaussian	25	1000	3871.3	3656.1	3647.6	106.2	164.5	156.6	3861.0	3707.1	3707.1	98.4	300	300
Clayton	5	200	1372.0	892.4 *	892.8 *	7.0	9.1	9.4	994.4	655.4 *	655.4 *	7.0	10	10
Clayton	5	1000	6775.3	4453.3 *	4453.3 *	7.0	10.0	10.0	4838.4	3228.7 *	3228.7 *	7.0	10	10
Clayton	25	200	23185.5	12843.5 *	12613.1 *	47.0	127.3	118.0	16329.6	9466.5 *	9466.5 *	47.0	300	300
Frank	5	200	385.4	285.0 *	284.3 *	7.0	9.1	9.2	351.1	259.9 *	259.9 *	7.0	10	10
Frank	5	1000	1852.9	1391.7 *	1384.9 *	7.0	10.0	10.0	1683.4	1275.7 *	1275.7 *	7.0	10	10
Gumbel	5	200	1940.7	1227.7 *	1199.6 *	7.0	9.3	9.6	1614.3	1018.9 *	1018.9 *	7.0	10	10
Gumbel	5	1000	8504.1	5522.0 *	5518.5 *	7.0	10.0	10.0	7949.7	5055.3 *	5055.3 *	7.0	10	10
Gumbel	25	200	34303.3	18616.2 *	18382.2 *	49.9	128.9	119.9	26137.7	14461.3 *	14461.3 *	47.0	300	300
hClayton	5	200	1090.9	708.8 *	708.5 *	7.0	8.5	8.5	795.4	530.1 *	530.1 *	7.0	10	10
hClayton	5	1000	5394.6	3523.8 *	3524.6 *	7.0	9.7	9.8	3880.6	2607.1 *	2607.1 *	7.0	10	10
hClayton	20	200	6139.0	3780.9 *	3734.0 *	37.0	84.9	74.9	4561.5	2968.2 *	2968.2 *	37.0	190	190
hClayton	25	200	7869.4	4857.7 *	4781.9 *	47.0	117.5	102.2	5873.5	3853.5 *	3853.5 *	47.0	300	300
hFrank	5	200	253.4	190.0 *	189.3 *	7.0	8.1	7.8	230.7	174.5 *	174.5 *	7.0	10	10
hFrank	5	1000	1236.9	935.0 *	934.5 *	7.0	9.7	9.8	1124.2	860.7 *	860.7 *	7.0	10	10
hFrank	10	1000	3169.8	2427.4 *	2415.7 *	17.0	39.0	39.2	2886.6	2252.7 *	2252.7 *	17.0	45	45
hFrank	20	200	1595.1	1204.0 *	1171.8 *	37.2	80.2	68.9	1470.1	1166.0 *	1166.0 *	37.0	190	190
hFrank	25	200	2074.7	1578.1 *	1527.2 *	47.3	111.0	93.4	1913.8	1550.3 *	1550.3 *	47.0	300	300
hGumbel	5	200	1490.5	1005.3 *	975.5 *	7.0	8.4	8.3	1215.4	780.1 *	780.1 *	7.0	10	10
hGumbel	5	1000	6443.9	4177.8 *	4177.2 *	7.0	9.5	9.7	5975.8	3849.9 *	3849.9 *	7.0	10	10
hGumbel	10	1000	16086.6	9957.4 *	9949.4 *	17.0	37.5	36.9	14953.0	9156.7 *	9156.7 *	17.0	45	45
hGumbel	20	200	8622.2	4858.3 *	4735.4 *	37.0	86.1	75.8	6868.5	4161.6 *	4161.6 *	37.0	190	190
hGumbel	25	200	11777.4	6444.8 *	6319.3 *	47.0	120.0	103.3	8782.5	5351.3 *	5351.3 *	47.0	300	300

Table 2: Simulated data. Average log-likelihood values over 350 simulation runs (larger is better) and the number of parameters of the estimated cDAG, C-vine and R-vine. The letter 'h' indicates that data are generated from a hierarchical copula. The '*' symbol indicates that based on a one-sided Mann-Witney test, the p-value for testing the null hypothesis of identical distributions, against the alternative 'the distribution for the quantity of interest coming from the cDAG model is shifted to the right of the distribution for the C-vine (R-vine) model', is lower than 5%. Due to multiple testing a Bonferroni correction has been applied to keep the familywise error rate at 95%.

SSQ_τ and SSQ_{Gini}. Averages across all 350 simulation runs are presented. A schematic overview of the simulation procedure is contained in Algorithm 3.

From Table 2 we observe that the cDAGs result in similar or larger log-likelihood values for a multitude of simulation settings. Even for settings where there is a fixed Gaussian copula there is a gain by using the cDAG approach, and as expected for all methods the log-likelihood values are increasing when the copula is selected rather than fixed to the Gaussian copula.

Out-of-sample prediction is used for Kendall's τ and Gini's index. A one-sided Mann-Witney test (with Bonferroni correction) indicates significant lower values for the cDAG as compared with the C- or R-vine, see Table 3. We have repeated the simulation study by using only the Gaussian copula. That is, the structure of the graph is estimated freely, but for all terms in the decomposition we use the Gaussian copula. For the C- and R vines, that meant using the Gaussian copula for all $p \times (p-1)/2$ terms. The results are presented in Table 3 and indicate that taking advantage of the local decompositions as in the cDAG can lead to improvements in the fit. This indicates the benefit of including the structure of the DAG in the modeling aspect. Similar conclusions (not reported here) have been reached inspecting Spearman's ρ and Blomqvist's β as dependence measures.

While the cDAG has the possibility of using more bivariate copulas than in a C-vine approach, the estimated number of parameters presented in Table 2 shows that this is not always the case. The range of values indicates that for some simulated settings a simple and sparsely estimated DAG suffices, whereas for other settings one needs to estimate a complex and dense structure in order to adequately capture the features of the data.

7.2 Simulation settings B

Instead of evaluating the log-likelihood on the same dataset as used to estimate the parameters, we also evaluate it on an external, independent 'hold-out' sample. The number of observations for the hold-out sample was taken equal to 1200. For the evaluation part (see Algorithm 3) we now simulate datasets of size $n_2 \times p$ where the number of simulated cases n_2 was set at 1000 and p is the number of nodes in the graph. One last modification to the approach we took so far, is to replace the sequential estimation of parameters (where the estimation of the copula parameters involved in the bivariate decompositions at the T-th tree depends on the parameters estimated at tree $T-1$) by a joint estimation of all parameters (where the full log-likelihood is numerically optimized, starting from suitable initial values). The latter, is computationally more involved than the sequential estimation scheme. Based on these three modifications to the approach in simulation settings A, we present in Table 4 the out-of-sample log-likelihood, the number of estimated parameters, SSQ_τ and SSQ_{Gini} when the full joint estimation of parameters is used. The number of nodes in the graph was $p = 5, 10, 15$ and the sample size n used in the estimation step was set at 1000 cases. We also present the results of a two-sided Mann-Witney test to test the null hypothesis of identical distributions, with a Bonferroni correction for multiple testing. This table leads to similar conclusions as Tables 2 and 3. In several instances the C- and R-vines seem to gain more in performance than the cDAGs. Most notably, there is a tendency for the vines to provide a higher out-of-sample log-likelihood value when the data are generated from a DAG with t4 or Gaussian errors with low sparsity. Moreover, when the data were generated from a Clayton copula the vines were also producing lower squared errors than the cDAG. The main message of the comparison is that none of the methods is everywhere the best, and that the joint estimation method, albeit more time consuming, can lead to better performances especially for the vine models.

8 Euro Stoxx 50 dataset

We use a financial dataset that contains the daily log-returns in the four-year period May 22, 2006 to April 29, 2010 for the Euro Stoxx 50 index, five national indices (DAX-Germany, CAC-France, IBEX-Spain, FTSE-UK and AEX-The Netherlands) and 45 stocks: 12 from the German financial market, 18 from the French market, 6 from the UK market, 5 from the Spanish market and 4 from the Dutch financial market. The dataset was constructed from financial data stored on the Yahoo Finance servers

			Estimated copula						Gaussian copula					
			SSQ_τ			SSQ_{Gini}			SSQ_τ			SSQ_{Gini}		
Data	p	n	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine
t4	5	200	0.064	0.126 *	0.133 *	0.051	0.094 *	0.100 *	0.077	0.125 *	0.131 *	0.061	0.098 *	0.103 *
t4	5	1000	0.061	0.100 *	0.110 *	0.047	0.074 *	0.083 *	0.073	0.119 *	0.129 *	0.056	0.090 *	0.098 *
t4	25	200	4.487	6.252 *	6.105 *	3.352	4.566 *	4.476 *	4.116	4.267	4.139	3.135	3.257 *	3.175
Gaussian	5	200	0.069	0.125 *	0.129 *	0.055	0.094 *	0.097 *	0.069	0.131 *	0.123 *	0.056	0.100 *	0.097 *
Gaussian	5	1000	0.060	0.112 *	0.111 *	0.047	0.084 *	0.084 *	0.062	0.122 *	0.116 *	0.048	0.091 *	0.090 *
Gaussian	25	200	1.552	2.008 *	2.208 *	1.132	1.443 *	1.566 *	1.003	1.089	1.038	0.742	0.806	0.778
Gaussian	25	1000	0.772	0.871 *	0.845	0.551	0.616 *	0.609 *	0.724	0.827 *	0.780	0.516	0.586 *	0.562 *
Clayton	5	200	0.024	0.042 *	0.047 *	0.010	0.017 *	0.020 *	0.033	0.038 *	0.068 *	0.019	0.022 *	0.041 *
Clayton	5	1000	0.019	0.037 *	0.041 *	0.006	0.013 *	0.016 *	0.030	0.032	0.067 *	0.019	0.021 *	0.044 *
Clayton	25	200	0.026	0.034 *	0.039	0.007	0.010 *	0.013 *	0.306	0.233	0.268	0.116	0.087	0.104
Frank	5	200	0.038	0.048 *	0.061 *	0.031	0.032	0.035	0.046	0.059 *	0.067 *	0.038	0.044 *	0.052 *
Frank	5	1000	0.035	0.046 *	0.061 *	0.028	0.030	0.033 *	0.040	0.055 *	0.060 *	0.034	0.042 *	0.048 *
Gumbel	5	200	0.009	0.014 *	0.024 *	0.004	0.006 *	0.010 *	0.006	0.013 *	0.017 *	0.003	0.005 *	0.008 *
Gumbel	5	1000	0.006	0.010 *	0.022 *	0.003	0.004 *	0.009 *	0.003	0.010 *	0.016 *	0.002	0.003 *	0.007 *
Gumbel	25	200	0.104	0.114 *	0.127 *	0.052	0.055 *	0.060 *	0.005	0.006 *	0.006 *	0.002	0.002 *	0.002 *
hClayton	5	200	0.030	0.053 *	0.068 *	0.020	0.027 *	0.030 *	0.031	0.047 *	0.066 *	0.021	0.029 *	0.041 *
hClayton	5	1000	0.023	0.049 *	0.059 *	0.016	0.024 *	0.024 *	0.025	0.042 *	0.061 *	0.019	0.027 *	0.040 *
hClayton	20	200	0.230	0.237	0.348 *	0.118	0.112	0.177 *	0.301	0.366 *	0.318	0.177	0.227 *	0.203 *
hClayton	25	200	0.298	0.348 *	0.355 *	0.150	0.172 *	0.184 *	0.372	0.322	0.431 *	0.230	0.198	0.271 *
hFrank	5	200	0.050	0.071 *	0.089 *	0.038	0.051 *	0.056 *	0.054	0.083 *	0.074 *	0.047	0.061 *	0.055 *
hFrank	5	1000	0.043	0.070 *	0.079 *	0.032	0.049 *	0.047 *	0.048	0.078 *	0.067 *	0.044	0.058 *	0.048
hFrank	10	1000	0.099	0.203 *	0.167 *	0.078	0.140 *	0.118 *	0.129	0.270 *	0.232 *	0.115	0.219 *	0.191 *
hFrank	20	200	0.329	0.521 *	0.726 *	0.234	0.365 *	0.497 *	0.474	0.519 *	0.510	0.394	0.425	0.404
hFrank	25	200	0.413	0.585 *	0.965 *	0.296	0.403 *	0.667 *	0.581	0.553	0.648 *	0.487	0.437	0.531 *
hGumbel	5	200	0.013	0.027 *	0.037 *	0.008	0.014 *	0.017 *	0.010	0.030 *	0.030 *	0.007	0.016 *	0.015 *
hGumbel	5	1000	0.010	0.023 *	0.031 *	0.006	0.011 *	0.014 *	0.006	0.027 *	0.028 *	0.005	0.014 *	0.013 *
hGumbel	10	1000	0.030	0.120 *	0.063 *	0.011	0.062 *	0.032 *	0.021	0.080 *	0.063 *	0.009	0.043 *	0.033 *
hGumbel	20	200	0.134	0.228 *	0.278 *	0.060	0.114 *	0.141 *	0.107	0.128 *	0.126 *	0.056	0.070 *	0.071 *
hGumbel	25	200	0.176	0.273 *	0.394 *	0.090	0.134 *	0.194 *	0.147	0.145	0.133	0.082	0.075	0.075

Table 3: Simulated data. Averages over 350 simulation runs of SSQs for Kendall's τ and Gini's index for the estimated cDAG, C-vine and R-vine; lower is better. The letter 'h' indicates that data are generated from a hierarchical copula. The '*' symbol indicates that based on a one-sided Mann-Witney test, the p-value for testing the null hypothesis of identical distributions, against the alternative 'the distribution for the quantity of interest coming from the cDAG model is shifted to the left of the distribution for the C-vine (R-vine) model', is lower than 5%. Due to multiple testing a Bonferroni correction has been applied to keep the familywise error rate at 95%.

		log-likelihood (Out-of-sample)			No. Parameters			SSQ_τ			SSQ_{Gini}		
Data	p	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine	cDAG	C-vine	R-vine
t4 (Sparsity .9)	5	15.3	19.1 *	19.1 *	14.3	1.3	1.3	.0091	.0190 *	.0205 *	.0074	.0137 *	.0157*
t4 (Sparsity .9)	10	85.8	98.4 *	98.6 *	85.2	6.6	6.5	.0560	.0971 *	.0930 *	.0448	.0731 *	.0696*
t4 (Sparsity .9)	15	202.8	225.4	226.6	209.9	15.8	15.4	.1282	.2187 *	.2169 *	.1025	.1636 *	.1627*
t4 (Sparsity .6)	5	99.6	95.2 *	95.2 *	11.3	4.3	4.1	.0083	.0164 *	.0190 *	.0066	.0121 *	.0146*
t4 (Sparsity .6)	10	479.7	443.5 *	444.4 *	41.9	21.5	19.8	.0286	.0512 *	.0483 *	.0231	.0397 *	.0377*
t4 (Sparsity .6)	15	1234.1	1145.8 *	1146.3 *	79.7	54.9	5.7	.0540	.0815 *	.0836 *	.0446	.0639 *	.0659*
Gaussian (Sparsity .9)	5	12.7	16.5 *	16.6 *	14.8	1.4	1.4	.0097	.0198 *	.0212 *	.0080	.0144 *	.0164*
Gaussian (Sparsity .9)	10	72.0	85.0 *	85.2 *	85.5	6.5	6.5	.0564	.0980 *	.0942 *	.0452	.0740 *	.0707*
Gaussian (Sparsity .9)	15	171.0	195.0 *	196.1	208.2	15.3	15.0	.1288	.2210 *	.2172 *	.1034	.1647 *	.1620*
Gaussian (Sparsity .6)	5	86.0	82.5 *	82.8 *	1.8	4.2	4.1	.0090	.0164 *	.0186 *	.0070	.0119 *	.0141*
Gaussian (Sparsity .6)	10	427.9	395.8 *	396.9 *	42.9	2.5	19.2	.0306	.0528 *	.0494 *	.0246	.0404 *	.0382*
Gaussian (Sparsity .6)	15	1119.8	1039.6 *	1039.5 *	82.5	52.4	48.2	.0575	.0846 *	.0829 *	.0469	.0655 *	.0639*
Clayton	5	5722.8	3852.8 *	3852.8 *	7.0	1.0	1.0	.0544	.0487 *	.0800 *	.0319	.0291	.0438*
Clayton	10	21087.5	13345.8 *	13344.1 *	17.0	44.7	44.9	.1737	.1626 *	.1680 *	.0777	.0709 *	.0767*
Clayton	15	4079.1	25087.4 *	2508.3 *	27.0	10.6	103.1	.2921	.2575 *	.2631 *	.1108	.0994 *	.1016
Frank	5	1956.8	1515.4 *	1515.4 *	7.0	1.0	1.0	.0248	.0353 *	.0628 *	.0278	.0377 *	.0564*
Frank	10	11296.1	7908.9 *	791.7 *	17.0	44.4	44.9	.1346	.1348	.1370	.1009	.0999 *	.1056*
Frank	15	25617.9	16997.1 *	17002.2 *	27.0	98.7	102.2	.2585	.2569 *	.2735 *	.1488	.1477 *	.1551*
Gumbel	5	9443.9	605.2 *	605.2 *	7.0	1.0	1.0	.0046	.0027 *	.0085 *	.0021	.0015	.0036*
Gumbel	10	3636.7	21488.1 *	21409.2 *	17.0	44.3	44.8	.0034	.0026 *	.0031 *	.0014	.0011 *	.0014*
Gumbel	15	70559.0	40205.2 *	39532.8 *	27.0	98.4	101.7	.0023	.0027 *	.0026 *	.0009	.0010 *	.0010*
hClayton	5	4598.4	3111.7 *	3111.7 *	7.0	1.0	1.0	.0381	.0379 *	.0685 *	.0244	.0238 *	.0406*
hClayton	10	11318.9	7549.2 *	7549.5 *	17.0	41.8	42.6	.1009	.1011 *	.1015 *	.0627	.0618 *	.0668*
hClayton	15	18051.4	12155.2 *	12152.8 *	27.0	89.9	91.5	.1603	.1619 *	.1721 *	.1028	.1045 *	.1091*
hFrank	5	1304.7	1014.5 *	1014.9 *	7.0	9.8	9.9	.0148	.0278 *	.0508 *	.0193	.0298 *	.0481*
hFrank	10	3274.0	264.4 *	2641.3 *	17.0	4.2	41.1	.0990	.1181 *	.1282 *	.1089	.1248	.1335*
hFrank	15	5256.7	4389.9 *	4383.9 *	27.0	85.0	86.7	.0506	.0739 *	.0669 *	.0583	.0774	.0730*
hGumbel	5	7113.1	4602.4 *	4602.7 *	7.0	9.7	9.8	.0045	.0033 *	.0140 *	.0024	.0020 *	.0070*
hGumbel	10	17629.7	10901.9 *	1090.7 *	17.0	38.7	38.9	.0113	.0101 *	.0114 *	.0061	.0055	.0065*

Table 4: Simulated data. Averages over 350 simulation runs of out-of-sample log-likelihood, number of bivariate terms, SSQs for Kendall's τ and Gini's index for the estimated cDAG, C-vine and R-vine. The letter 'h' indicates that data are generated from a hierarchical copula. The '*' symbol indicates that based on a two-sided Mann-Witney test, the p-value for testing the null hypothesis of identical distributions, is lower than 5%. Due to multiple testing a Bonferroni correction has been applied to keep the familywise error rate at 95%.

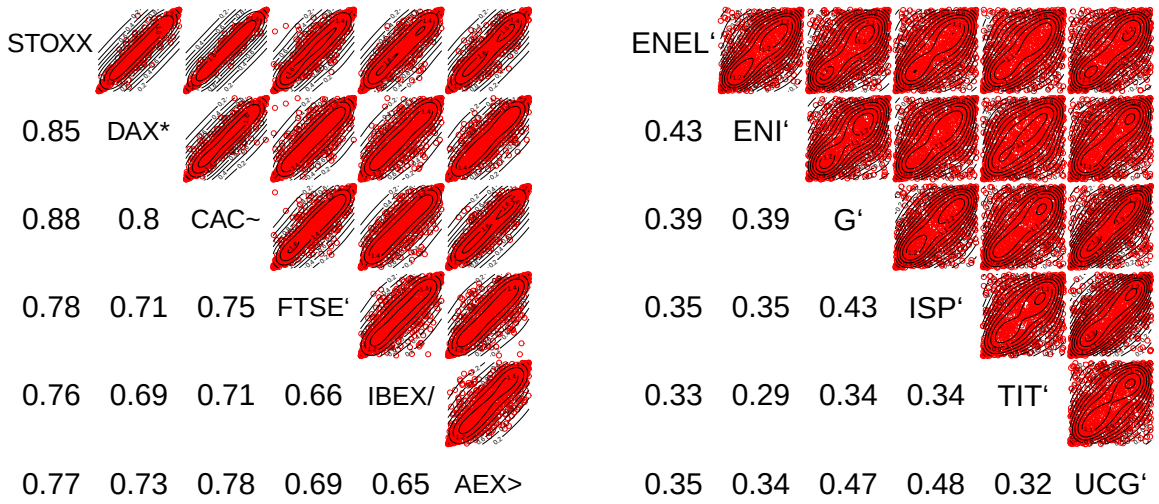


Fig. 8: Euro Stoxx 50 data. Pairwise scatterplots and contour lines of empirical copulas for the five national indices and the Euro Stoxx 50 index (left panel) and for individual stocks from the FTSE index (right panel). The numbers below the main diagonal represent the observed Kendall's τ measure of association.

(http://finance.yahoo.com/). We refer to Brechmann and Czado (2013) for a description of the dataset (which contained data on one extra stock from the French financial market, for which we have not found publicly available data) and for an application of several vine-based financial models. We have first fitted an ARMA(1,1)-Garch(1,2) for each times series, retained the residual observations from which we have then constructed the pseudo-observations, a vector of 965 data points for each series. In all calculations, we have estimated models using the pseudo-observations as input data.

In Figure 8 we present pairwise scatterplots of the five national indices and the European index using the pseudo-observations (left panel), as well as scatterplots for the stocks from the UK market (right panel). For each variable the marginal distributions seem close to uniform distributions, with all data between 0 and 1. Therefore, plots of the marginal distributions and the axes have been omitted for ease of presentation. Contour lines corresponding to the bivariate empirical copula with uniform margins have been superimposed in each scatterplot. Below the main diagonal we present the observed Kendall's τ values for each pair of variable. The conclusion of the figure is that for some pairs of nodes a bivariate Gaussian copula model might be less appropriate than for others. The scatterplots of the national indices seem to show a strong dependence, larger than that shown by the stocks from the UK market and some of the pairs show signs of bimodality with respect to the empirical distributions.

We compare the cDAG estimated model in Figure 9 to the tree structure of an R-vine in Brechmann and Czado (2013). We have redrawn Figure 4 from their article using the same layout as for the cDAG, see Figure 9(b). We observe that both the cDAG approach, see Figure 9(a), as well as the first level tree from an R-vine procedure as implemented in the above work, place the Euro Stoxx 50 index as a central node linked to the five national indices. Both approaches illustrate nicely the expected links between the national indices and the individual stocks from the five countries. A bonus of the cDAG is the added information contained in the DAG, namely, first, within a country index there seems to be a quite high level of stock interaction as the log returns of certain stocks influence the log-returns of others. Second, at the European level the national indices interact with each other.

In order to list which conditional independencies can be read from the graph, we have applied the 'd-separation' criterion (Geiger et al, 1990) to the estimated cDAG according to which, if two variables are d-separated relative to a set of variables Z, then they are also independent conditionally on Z (see the Appendix for a definition of d-separation). The analysis indicates that conditioning on the general Euro Stoxx 50 leaves any pair of national indices still conditionally dependent.

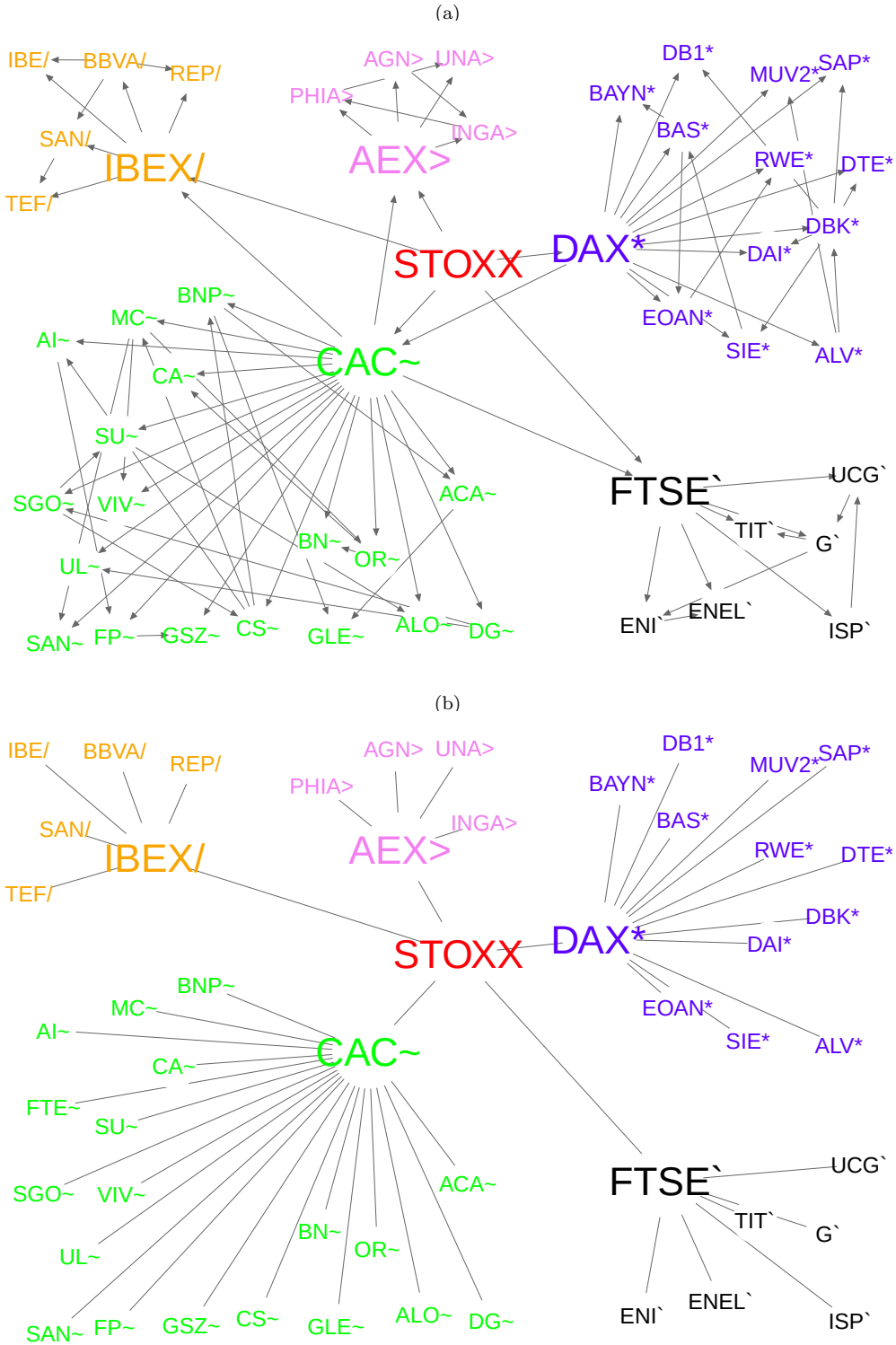


Fig. 9: Euro Stoxx 50 data. (a) Estimated cDAG using the cDAG-IC criterion and (b) the first tree of an R-vine model redrawn from Figure 4 from Brechmann and Czado (2013) using the same layout as in (a).

A first interesting observation resulting from the application of the d-separation criterion, is that conditioning on the evolution of the general Euro Stoxx 50 index and on the French CAC index, makes any couple of the remaining national indices independent, highlighting that the French market played a central role in terms of log-returns among these markets in the analyzed period. Second, when focusing on the French market we observe that conditioning only on the national index leads to having any couple of stock conditionally dependent, showing that in the French financial market there is more heterogeneity in the log-returns than the index can capture. The same holds for the four other markets in this analysis.

Considering stocks from different sectors, we can for example, conclude that conditioned upon the log-returns of stocks from the bank sector such as BNP Paribas (BNP), Credit Agricole (ACA) or AXA Group (CS) makes the evolution of the log-returns for Société Générale (GLE) independent of the evolution of stocks of a different sector, for example, L'Oreal (OR), Carrefour (CA) or Louis Vuitton (MC).

As explained in Section 5.4, for an across-class comparison we now compare via AIC, BIC and Vuong's hypothesis test (see Vuong, 1989) the estimated cDAG against 19 other models which include: several C- and R-vine models (estimated using either AIC or BIC and with or without independence tests between nodes), several Bayesian networks estimated using the BGe/AIC/BIC score in conjunction with a hill-climbing (HC) procedure (as implemented in Scutari, 2010), the estimated Bayesian networks using the PC (as implemented in Kalisch et al, 2012) or SIN algorithm (as implemented in Drton and Perlman, 2008) in both cases using α=0.05 or α=0.1 and the Bayesian network estimated using the nonparametric Bayesian belief net (NPBBN) described in Hanea et al (2010) and Hanea (2011). In the case of NPBBN for the estimation of the DAG, we have used the software provided by the authors. We started from a fully connected graph and eliminated edges for which the empirical correlation values were low, until a sparse graph was obtained that passed all offered tests in that software. Afterwards, the graph structure has been mapped to a D-vine in the R software using the CDVine package, where we used a Gaussian copula to fit certain edges associated with conditional correlation coefficients and independence copulas to alleviate all unnecessary terms.

Except for the cDAG, the C-vines, the R-vines and the NPBBN method, all other models make the explicit assumption of multivariate normality which is likely to be violated in this case. We compute the AIC and BIC values (smaller values indicate better performance) for each estimated graph as $\text{AIC} = -2\text{log-Lik}(\hat{\boldsymbol{\theta}}; G) + 2\text{length}(\hat{\boldsymbol{\theta}})$ and $\text{BIC} = -2\text{log-Lik}(\hat{\boldsymbol{\theta}}; G) + \log(965)\text{length}(\hat{\boldsymbol{\theta}})$. Table 5 shows the in-sample and the 3-fold crossvalidated log-likelihood values, the number of estimated parameters, the AIC and BIC values for each model and the number of seconds the estimation process took. For the C-vine and R-vine approaches, AIC and BIC happened to select the same best model.

Out of tested models the best BIC value was obtained by the cDAG, which provided a log-likelihood value 1.36 times higher than the second best model, namely the C-vine specification where we use BIC (with the same selection as AIC) for model selection and allow for model simplifications by using independence testing, and 2.11 times higher than the best performing Bayesian network assuming multivariate normality. When inspecting the 3-fold cross-validated log-likelihood, the C-vine method with independence testing was second best.

Testing each competitor model against the cDAG model using either adjusted (based on AIC or BIC penalties) or unadjusted versions of the Vuong tests (where we apply a Bonferoni correction for the multiple testing issue) leads to the conclusion that the cDAG is the preferred model. Allowing flexibility in the distribution improves the fit, and coupling distributional flexibility with the DAG structure improves the fit even more.

One simple way to try to apply methods requiring a normality assumption to non-normal data is to use a transformed version of the pseudo-observations, where at each node we apply the normal quantile function to the pseudo-observations. Such a transformation has the explicit purpose of making each marginal distribution closer to a normal distribution. This marginal near-normality, however, is not sufficient. Estimating a DAG using this transformed dataset resulted in very similar log-likelihood values as compared to using the untransformed data. This holds for both the in-sample and the cross-validated case. See Table 5 for a comparison.

Model	Seconds	log-Lik	Par.	AIC	BIC	log-Lik(CV)
cDAG	1525.78	38206.9 (1)	94	-76225.9 (1)	-75767.9 (1)	37531.4 (1)
C-vine, BIC&AIC, Indep Test	502.68	28082.1	458	-55248.2	-53016.8 (2)	25974.1 (2)
R-vine, BIC&AIC, Indep Test	217.00	27979.1	434	-55090.2	-52975.7	25950.9
C-vine, BIC&AIC, No Indep Test	660.96	29016.5 (2)	1275	-55483.0 (2)	-49271.0	25725.6
R-vine, BIC&AIC, No Indep Test	220.95	28908.9	1275	-55267.9	-49055.9	25706.2
NPBBN	124.13	15904.6	253	-31303.2	-30070.6	15400.9
HC, BGe, No Transf.	37.35	18291.2	331	-35920.5	-34307.8	18123.1
HC, BIC, No Transf.	2.89	18102.0	271	-35662.1	-34341.8	17961.1
HC, AIC, No Transf.	10.44	18669.2	540	-36258.4	-33627.5	18671.5
PC, $\alpha = .1$, No Transf.	64.92	12942.0	199	-25486.0	-24516.4	12699.0
PC, $\alpha = .05$, No Transf.	39.43	13715.4	187	-27056.7	-26145.6	13173.2
SIN, $\alpha = .1$, No Transf.	0.14	16350.4	158	-32384.8	-31615.0	15137.4
SIN, $\alpha = .05$, No Transf.	0.11	16043.8	149	-31789.6	-31063.7	14712.8
HC, BGe, Normal Transf.	43.96	18243.5	339	-35809.1	-34157.4	18116.7
HC, BIC, Normal Transf.	4.35	18072.0	283	-35578.0	-34199.2	17898.5
HC, AIC, Normal Transf.	15.17	18595.8	557	-36077.6	-33363.8	18679.2
PC, $\alpha = .1$, Normal Transf.	102.0	13113.9	194	-25839.9	-24894.7	12419.6
PC, $\alpha = .05$, Normal Transf.	48.6	13324.4	184	-26280.9	-25384.4	12829.3
SIN, $\alpha = .1$, Normal Transf.	0.29	16577.1	169	-32816.1	-31992.8	14940.1
SIN, $\alpha = .05$, Normal Transf.	0.12	16097.4	159	-31876.7	-31102.1	14712.8

Table 5: Euro Stoxx 50 data. Summary measures for 20 estimated models. Ranks (1) and (2) indicate the first two best scoring models. The columns 'log-Lik' and 'log-Lik (CV)' refer to the log-likelihood values obtained when using, respectively, the entire sample and the values obtained using 3-fold cross-validation. 'Par.' refers to the number of parameters estimated by the method.

9 Discussion

For computational reasons the current approach uses one copula family per $\frac{CV_l}{DV_l}$ ratio at a node l, although for different nodes we allow for different families. With an increased computational cost one can allow different copula families in both the numerator and denominator as our method is directly extendable to deal with such cases. Additionally, different ordering of the parents could be used in the numerator and denominator and as such a quick way for performing a rough sensitivity analysis could be obtained.

The current method is a fully parametric method where one selects copula families from a predefined list of relevant copulas for the problem at hand. At this stage, we did not allow the copula parameter to depend on covariate values, as all copula models used here make this simplifying assumption. An interesting extension could be to introduce such functional dependence where the parameters may depend on the covariate values as implied by the conditioning set, resulting in a semiparametric model where smoothing methods such as local polynomial estimation, regression splines, etc. could be investigated. Working fully nonparametrically would introduce even more flexibility in the modeling process, by estimating nonparametrically appropriate copulas at each of the nodes.

Instead of using classical DAGs, conceptually our method could be extended towards using chain graphs (that contain combinations of directed and undirected edges) or graphs that account for a temporal or ordering aspect of the data by using hidden Markov models or by introducing time dependence and allowing the parameter θ to functionally depend on the time. Local likelihoods could be introduced to model such time dependence. Such extensions and open problems are subject to future research.

Acknowledgements

We wish to thank the reviewers for their comments. We thank A. Hanea and D. Ababei for providing the software of their procedure. We acknowledge the support of the Fund for Scientific Research Flanders, KU Leuven grant GOA/12/14 and of the IAP Research Network P7/06 of the Belgian Science Policy. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Hercules Foundation and the Flemish Government - department EWI.

Appendix A: Technical details

Assumptions of Proposition 4.1 adapted from (Sin and White, 1996). For every node l in the graph, let $q_{lk}(\cdot,\boldsymbol{\theta}) = \log CV_{l,k}(\boldsymbol{\theta}_{CV_l}) - \log DV_{l,k}(\boldsymbol{\theta}_{DV_l})$, $\tilde{q}_{lk}(\cdot,\boldsymbol{\theta}) = \log DV_{l,k}(\boldsymbol{\theta}_{DV_l})$ and log-Lik$(\cdot,\boldsymbol{\theta};\text{node}_l) \equiv Q_{ln}(\cdot,\boldsymbol{\theta}) = \sum_{k=1}^{n} q_{lk}(\cdot,\boldsymbol{\theta})$ with $\boldsymbol{\theta} = (\boldsymbol{\theta}_{CV_l},\boldsymbol{\theta}_{DV_l})$ and $k = 1,2,\ldots,n$. For ease of exposition we state general conditions that need to be satisfied by $q_{lk}(\cdot,\boldsymbol{\theta})$, $\tilde{q}_{lk}(\cdot,\boldsymbol{\theta})$, $Q_{ln}(\cdot,\boldsymbol{\theta})$ and $\boldsymbol{\theta}$ for every model m.

Let $(\Theta,\mathcal{F},P)$ be a complete probability space and Θ be a compact subset of $\mathbb{R}^d$ with $d \in \mathbb{N}$. For all $n \in \mathbb{N}$ let $Q_{ln} : \Omega \times \Theta \to \mathbb{R}$ be such that:

i $\forall \boldsymbol{\theta} \in \Theta$, $Q_{ln}(\cdot,\boldsymbol{\theta})$ is $\mathcal{F}$-measurable.
ii $\forall \omega \in A \in \mathcal{F}$ with $P(A) = 1$, $Q_{ln}(\omega,\cdot)$ is continuously differentiable on Θ.
iii The expectation $E(Q_{ln}(\cdot,\boldsymbol{\theta}))$ exists and defines a function which is continuously differentiable on Θ and $\bigtriangledown E(Q_{ln}(\cdot,\boldsymbol{\theta})) = E(\bigtriangledown Q_{ln}(\cdot,\boldsymbol{\theta}))$ where $\bigtriangledown$ is the gradient operator.
iv The least false parameter $\boldsymbol{\theta}_{0n} = \arg\sup_{\boldsymbol{\theta}\in\Theta} \frac{1}{n} E(Q_{ln}(\cdot,\boldsymbol{\theta}))$ is interior to Θ uniformly (in n).
v Given $\epsilon > 0$ there exists $N_0(\epsilon) < \infty$ and $\delta(\epsilon) > 0$ such that $\inf\{\min\{K_n^*(\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathcal{N}_n^*(\epsilon)^c\}, n > N_0(\epsilon)\} \equiv \delta(\epsilon)$, where $K_n^*(\boldsymbol{\theta}) \equiv n^{-1}E(Q_{ln}(\cdot,\boldsymbol{\theta}_{0n})) - n^{-1}E(Q_{ln}(\cdot,\boldsymbol{\theta}))$, $\mathcal{N}_n^*(\epsilon)^c$ is the compact complement of $\mathcal{N}_n^*(\epsilon) \equiv \mathcal{S}_n^*(\epsilon) \cap \Theta$ in Θ and $\mathcal{S}_n^*(\epsilon)$ is an open sphere centered at $\boldsymbol{\theta}_{0n}$ with fixed radius ϵ.
vi For P-almost all ω, $q_{lk}(\omega,\cdot)$ is twice continuously differentiable as a function of $\boldsymbol{\theta}$, for $k = 1,2,\ldots$
vii q_{lk} and $\tilde{q}_{lk}$ satisfy a uniform weak law of large numbers (UWLLN) on Θ.
viii Each element of $\bigtriangledown q_{lk}(\cdot,\boldsymbol{\theta}_{0n})$ satisfies a central limit theorem.
ix $\exists \epsilon,\alpha > 0$ such that for P-almost all ω and for all n sufficiently large and for all $\boldsymbol{\theta} \in \mathcal{N}_n^*(\epsilon)$, $\det(n^{-1}\bigtriangledown^2 Q_{ln}(\omega,\boldsymbol{\theta})) \geq \alpha$, with $\mathcal{N}_n^*(\epsilon)$ as in Asumption v.
x For all n sufficiently large and for all $\boldsymbol{\theta} \in \mathcal{N}_n^*(\epsilon), E[n^{-1}\bigtriangledown^2 Q_{ln}(\cdot,\boldsymbol{\theta})]$ is $\boldsymbol{O}(1)$.
xi Each element of $\bigtriangledown^2 q_{ln}$ satisfies a UWLLN on $\mathcal{N}_n^*(\epsilon)$.

We assume that the copula densities are such that the above conditions are satisfied. These are basic assumptions that guarantee that $\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_{0n} = \boldsymbol{O}_p(n^{-1/2})$ and $Q_n(\cdot,\hat{\boldsymbol{\theta}}_n) - Q_n(\cdot,\boldsymbol{\theta}_{0n}) = \boldsymbol{O}_p(1)$. The asymptotic normality of $\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_{0n})$ for the models we consider has been shown in Hobæk Haff (2013).

Penalty conditions in Lemma 4 for the penalty in cDAG-IC

Proof Define $\Delta\widehat{\text{pen}}_{\text{cDAG}} = \widehat{\text{pen}}^1_{\text{cDAG}}(n,\hat{\boldsymbol{\theta}}^1) - \widehat{\text{pen}}^2_{\text{cDAG}}(n,\hat{\boldsymbol{\theta}}^2)$. For (i) it holds that

$$\Delta\widehat{\text{pen}}_{\text{cDAG}}/n = \Big(\frac{E\log DV_l^1}{|pa^1(l)|} - \frac{E\log DV_l^2}{|pa^2(l)|}\Big)\frac{1}{\log n} + o_P(1) = o_P(1).$$

The first equality holds due to Assumption vii.

For (ii) and (iii) it follows that

$$\frac{\Delta\widehat{\text{pen}}_{\text{cDAG}}}{\sqrt{n}} = \Big(\frac{E\log DV_l^1}{|pa^1(l)|} - \frac{E\log DV_l^2}{|pa^2(l)|}\Big)\frac{\sqrt{n}}{\log n} + o_P(\sqrt{n}),$$
$$\Delta\widehat{\text{pen}}_{\text{cDAG}} = \Big(\frac{E\log DV_l^1}{|pa^1(l)|} - \frac{E\log DV_l^2}{|pa^2(l)|}\Big)\frac{n}{\log n} + o_P(n).$$

By the assumed positiveness of the penalty difference, the conditions hold.

Definition of 'd-separation' between $\mathcal{X}$ and $\mathcal{Y}$ by $\mathcal{Z}$ (Barber, 2012). For every node $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, check every path $\mathcal{U}$ between x and y (that is, a sequence of nodes that starts in x and by following the directionality of the arrows leads to y). A path $\mathcal{U}$ is blocked if there is a node w in $\mathcal{U}$ such that either: (i) w is a collider (a collider node has two incoming arrows to it) and neither w nor any of its descendants is in $\mathcal{Z}$, or (ii) w is not a collider on $\mathcal{U}$ and w is in $\mathcal{Z}$. If all such paths are blocked then the sets of nodes $\mathcal{X}$ and $\mathcal{Y}$ are d-separated by $\mathcal{Z}$. If the sets of nodes $\mathcal{X}$ and $\mathcal{Y}$ are d-separated by $\mathcal{Z}$, they are independent conditional on $\mathcal{Z}$.

References

Aas K, Czado C, Frigessi A, Bakken H (2009) Pair-copula constructions of multiple dependence. Insurance: Mathematics and Economics 44(2):182–198

Akaike H (1973) Information theory and an extension of the maximum likelihood principle. In: Petrov B, Csáki F (eds) Second International Symposium on Information Theory, Akadémiai Kiadó, Budapest, pp 267–281

Bache K, Lichman M (2013) UCI machine learning repository

Barber D (2012) Bayesian Reasoning and Machine Learning. Cambridge University Press

Bauer A, Czado C, Klein T (2012) Pair-copula constructions for non-Gaussian DAG models. Canadian Journal of Statistics 40(1):86–109

Bedford T, Cooke RM (2001) Probability density decomposition for conditionally dependent random variables modeled by vines. Annals of Mathematics and Artificial Intelligence 32(1-4):245–268

Bedford T, Cooke RM (2002) Vines - a new graphical model for dependent random variables. The Annals of Statistics 30(4):1031–1068

Brechmann E, Czado C (2013) Risk management with high-dimensional vine copulas: An analysis of the Euro Stoxx 50. Statistics & Risk Modeling 30(4):307–342

Brechmann E, Schepsmeier U (2013) Modeling dependence with C- and D-vine copulas: The R package CDVine. Journal of Statistical Software 52(3):1–27

Brechmann EC, Czado C, Aas K (2012) Truncated regular vines in high dimensions with applications to financial data. Canadian Journal of Statistics 40(1):68–85

Chickering D (2002) Optimal structure identification with greedy search. Journal of Machine Learning Research 3:507–554

Clarke K (2003) Nonparametric model discrimination in international relations. Journal of Conflict Resolution 47(1):72–93

Cox D, Wermuth N (1996) Multivariate Dependencies: Models, Analysis and Interpretation. Chapman & Hall/CRC

Czado C (2010) Pair-copula constructions of multivariate copulas. In: Jaworki P, Durante F, Härdle W, Rychlik W (eds) Copula Theory and its Applications, Springer, pp 93–109

Czado C, Gärtner F, Min A (2011) Analysis of Australian electricity loads using joint Bayesian inference of D-vines with autoregressive margins. In: Kurowicka D, Joe H (eds) Dependence Modeling: Vine Copula Handbook, World Scientific Publishing, pp 265–280

Czado C, Schepsmeier U, Min A (2012) Maximum likelihood estimation of mixed C-vines with application to exchange rates. Statistical Modelling 12(3):229–255

Dißmann J, Brechmann E, Czado C, Kurowicka D (2013) Selecting and estimating regular vine copulae and application to financial returns. Computational Statistics & Data Analysis 59:52–69

Drton M, Perlman M (2008) A SINful approach to Gaussian graphical model selection. Journal of Statistical Planning and Inference 138(4):1179–1200

Elidan G (2010) Copula Bayesian networks. In: Lafferty J, Williams CKI, Shawe-Taylor J, Zemel R, Culotta A (eds) Advances in Neural Information Processing Systems 23, (NIPS 2010), pp 559–567

Elidan G (2012) Lightning-speed structure learning of nonlinear continuous networks. Journal of Machine Learning Research - Proceedings Track 22:355–363

Geiger D, Verma T, Pearl J (1990) Identifying independence in Bayesian networks. Networks 20(5):507–534

Genest C, Favre A (2007) Everything you always wanted to know about copula modeling but were afraid to ask. Journal of Hydrologic Engineering 12(4):347–368

Gijbels I, Veraverbeke N, Omelka M (2011) Conditional copulas, association measures and their applications. Computational Statistics & Data Analysis 55(5):1919–1932

Hanea AM (2011) Non-parameteric bayesian belief nets versus vines. In: Kurowicka D, Joe H (eds) Dependence Modeling: Vine Copula Handbook, World Scientific Publishing, pp 281–303

Hanea AM, Kurowicka D, Cooke RM, Ababei DA (2010) Mining and visualising ordinal data with non-parametric continuous BBNs. Computational Statistics & Data Analysis 54(3):668–687

Harris N, Drton M (2013) PC algorithm for nonparanormal graphical models. Journal of Machine Learning Research 14:3365–3383

Heckerman D, Geiger D (1995) Learning Bayesian networks: A unification for discrete and Gaussian domains. In: Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, pp 274–284

Hobæk Haff I (2013) Parameter estimation for pair-copula constructions. Bernoulli 19(2):462–491

Hofert M, Kojadinovic I, Maechler M, Yan J (2014) copula: Multivariate dependence with copulas. R package version 0.999-10.

Jalali A, Ravikumar P, Vasuki V, Sanghavi S (2010) On learning discrete graphical models using group-sparse regularization. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics

Joe H (1996) Families of m-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In: Rüschendorf L, Schweizer B, Taylor M (eds) Distributions with Fixed Marginals and Related Topics, Lecture Notes-Monograph Series, vol 28, Institute of Mathematical Statistics, pp 120–141

Kalisch M, Bühlmann P (2007) High-dimensional directed acyclic graphs with the PC-algorithm. Journal of Machine Learning Research 8:613–636

Kalisch M, Mächler M, Colombo D, Maathuis MH, Bühlmann P (2012) Causal inference using graphical models with the R package pcalg. Journal of Statistical Software 47(11):1–26

Koller D, Friedman N (2009) Probabilistic Graphical Models: Principles and Techniques. MIT Press

Kurowicka D, Cooke R (2002) The vine copula method for representing high dimensional dependent distributions: Applications to continuous belief nets. In: Yücesan E, Chen CH, Snowdon JL, Chames JM (eds) The Winter Simulation Conference, pp 270–278

Kurowicka D, Cooke R (2006) Uncertainty analysis with high dimensional dependence modelling. Wiley

Lauritzen S (1996) Graphical Models. Oxford University Press

Lee J, Hastie T (2014) Learning the structure of mixed graphical models. Journal of Computational and Graphical Statistics (In press)

Liu H, Lafferty J, Wasserman L (2009) The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. Journal of Machine Learning Research 10:2295–2328

Loh PL, Wainwright MJ (2013) Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. The Annals of Statistics 41(6):3022–3049

Lucas PJ (2007) Biomedical applications of Bayesian networks. In: Lucas PJF, Gámez J, Salmerón Cerdan A (eds) Advances in Probabilistic Graphical Models, Studies in Fuzziness and Soft Computing, vol 214, Springer, pp 333–358

Madsen AL, Kjærulff UB (2007) Applications of HUGIN to diagnosis and control of autonomous vehicles. In: Lucas PJF, Gámez J, Salmerón Cerdan A (eds) Advances in Probabilistic Graphical Models, Studies in Fuzziness and Soft Computing, vol 214, Springer, pp 313–332

Mari D, Kotz S (2001) Correlation and Dependence. Imperial College Press

Min A, Czado C (2010) Bayesian model selection for multivariate copulas using pair-copula constructions. Journal of Financial Econometrics 8(4):511–546

Min A, Czado C (2011) Bayesian model selection for D-vine pair-copula constructions. Canadian Journal of Statistics 39(2):239–258

Morales Nápoles O (2010) Bayesian belief nets and vines in aviation safety and other applications. PhD thesis, Technische Universiteit Delft

Nelsen RB (2006) An introduction to copulas. Springer

Okhrin O, Ristig A (2014) Hierarchical Archimedean copulae: The HAC package. Journal of Statistical Software 58(4):1–20

Peshkin L, Pfefer A, Savova V (2003) Bayesian nets in syntactic categorization of novel words. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Association for Computational Linguistics, vol 2, pp 79–81

R Core Team (2014) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria

Schepsmeier U, Stoeber J, Brechmann EC, Graeler B (2014) VineCopula: Statistical inference of vine copulas. R package version 1.3

Schwarz G (1978) Estimating the dimension of a model. The Annals of Statistics 6(2):461–464

Scutari M (2010) Learning bayesian networks with the bnlearn R package. Journal of Statistical Software 35(3):1–22

Sin C, White H (1996) Information criteria for selecting possibly misspecified parametric models. Journal of Econometrics 71(1-2):207–225

Sklar A (1959) Fonctions de répartition à n dimensions et leurs marges. Publ Inst Statist Univ Paris 8 pp 229–231

Smith M, Min A, Almeida C, Czado C (2010) Modeling longitudinal data using a pair-copula construction decomposition of serial dependence. Journal of the American Statistical Association 105:1467–1479

Spirtes P, Glymour C, Scheines R (2000) Causation, Prediction and Search, 2nd edn. MIT Press, Cambridge, MA

Vuong QH (1989) Likelihood ratio tests for model selection and non-nested hypotheses. Econometrica 57(2):307–333

Wainwright MJ, Jordan MI (2008) Graphical models, exponential families, and variational inference. Foundations and Trends in Machine Learning 1(1-2):1–305

Yang E, Ravikumar PK, Allen GI, Liu Z (2012) Graphical models via generalized linear models. In: Bartlett P, Pereira F, Burges C, Bottou L, Weinberger K (eds) Advances in Neural Information Processing Systems, (NIPS 2012), pp 1367–1375